

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.



PATENT ABSTRACTS OF JAPAN

(11) Publication number: 09081444 A

(43) Date of publication of application: 28.03.97

(51) Int. Cl.

G06F 12/00
G06F 12/00
G06F 17/30
H04N 5/78
H04N 7/173

(21) Application number: 07240351

(22) Date of filing: 19.09.95

(71) Applicant: FUJITSU LTD

(72) Inventor:
KATOU KOUKI
KONDOU AKIKO
ISHIKAWA HIROSHI

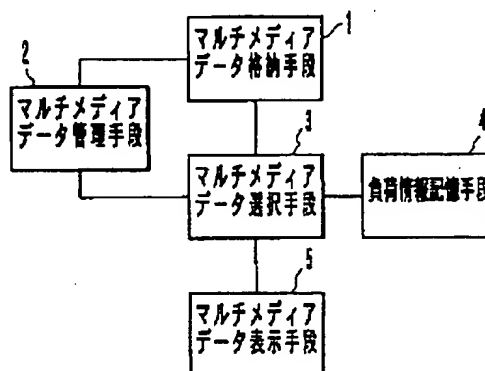
(54) MULTIMEDIA DATA RETRIEVAL SYSTEM,
MULTIMEDIA DATA SCHEDULING DEVICE,
MULTIMEDIA DATA REPRODUCING METHOD
AND REPRODUCING PROGRAM STORAGE
MEDIUM

COPYRIGHT: (C)1997,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a multimedia data retrieval system which is capable of controlling the load necessary of reproducing multimedia data.

SOLUTION: When a multimedia data management means 2 independently performs the management of the contents of describing the matter regarding an event and the management of the multimedia data which is corresponding to the contents and shows the substance of the contents, makes the plural multimedia data stored in a multimedia data storage means 1 correspond to one content and retrieved multimedia data is presented to a user, the load when it is displayed on a multimedia data display means 5 is read from a load information storage means 4 and multimedia data to be used for a presentation is selected from the plural multimedia data by a multimedia data selection means 3.



Our Ref: OP1169-US

(Prior Art Reference)

Japanese Patent Laid-Open Publication No. Hei 9-81444

Laid-Open Date: March 28, 1997

Filing No. Hei 7-240351

Filing Date: September 19, 1995

Applicant: 000005223

FUJITSU LIMITED

Kawasaki-shi, Kanagawa-ken, Japan

Inventors: Kohki KATO, Akiko KONDO and Hiroshi ISHIKAWA

all c/o FUJITSU LIMITED

Kawasaki-shi, Kanagawa-ken, Japan

[Title of the Invention]

MULTI-MEDIA DATA RETRIEVING SYSTEM, SCHEDULING DEVICE FOR
MULTI-MEDIA DATA, METHOD FOR REPRODUCING MULTI-MEDIA DATA AND
REPRODUCTION PROGRAM STORING MEDIUM

[Abstract]

[Object]

An object of the present invention is to provide a multi-media retrieving system capable of controlling a load required for reproducing multi-media data.

[Means for solving the problem]

A multi-media data managing means 2 manages contents describing the matters relating to an event and multi-media data corresponding to the contents and representing substance thereof, independently, associates a plurality of multi-media data stored in a multi-media data storing means 1 with one content, and in the case where the retrieved multi-media data is presented to a user, it reads out a load in displaying the data in a multi-media data displaying means 5 from a load information storing means 4, and selects the multi-media data to be used for presentation among the plurality of multi-media data by a multi-media data selecting means 3.

[0040]

[Embodiments of the Invention]

The multi-media data retrieving system of an embodiment of the present invention will be described hereinafter with reference to the accompanying drawings.

[0041]

In Fig. 1, multi-media data storing means 1 is for storing a plurality of multi-media data corresponding to one event. Multi-media data managing means 2 is for associating a common identifier representing the event with the plurality of multi-media data. Multi-media data selecting means 3 is for selecting one multi-media data among the plurality of multi-media data having the common identifier, according to a quantity of load required for a display. Load information storing means 4 is for associating the quantity of load required for processes in displaying information of the multi-media data with each of the plurality of multi-media data and storing it. Multi-media data displaying means 5 is for displaying multi-media data information.

[0042]

The multi-media managing means 2 manages the contents describing matters relating to the event, and the multi-media data corresponding to the contents and representing a substance thereof, independently, thereby the plurality of multi-media data can be associated with one content. When the retrieved multi-media data is presented to a user, the multi-media data to be used for presentation is selected among the plurality of multi-media data by referring to the quantity of load stored in the load information storing means 4, while considering computer resources.

[0043]

Fig. 2 is a block diagram showing a configuration of the multi-media data retrieving system of one embodiment of the present invention. In Fig. 2, reference numeral 11 denotes a CPU, wherein a data base engine 26 and an OS 27 are operated. Reference numeral 12 denotes a storage unit, wherein an OB data 13 managed by a data base and media data 20, if necessary, are stored. Reference numeral 13 denotes an object base (OB), storing a class information 14 and a method 15. The class information 14 has an instance for managing contents 16 and an instance for managing a stream 17.

[0044]

The method 15 comprises a scheduler 18, a reproduction program 19, a keyword () method 28, a setView () method 29, newVideo () method 30, a newAudio () method 31, a newRate () method 32, a newSize () method 33 and a play () method 34, all of which are used for reproducing the media data 20. The media data 20

is used in the case of using an external management mechanism of the data base, and storing the multi-media data. Reference numeral 21 is a memory. Reference numeral 22 is an I/O, which is connected to the storage unit 12, a keyboard 23 for inputting data, a CRT 24 for outputting the data, a speaker 25 and the CPU 11.

[0045]

As shown in Fig. 3, a window system 35 is managed by the CPU 11, and the CPU stores a MPEG-1 video data 36 and a MPEG-1 audio data 37 as the media data.

[0046]

Further, in the present embodiment, an object oriented data base (OODB) is used as the data base, and capability of dealing with multi-media is expanded on the basis of the object oriented data base.

[0047]

The multi-media data base is based on the object oriented data base, and manages the objects with hierarchical classification. A contents class representing the event and a stream class managing data for actually realizing the contents are provided. As shown in Fig. 4, as an instance of the contents class, there is existed a contents object 38 representing a specific event, and as instances of the stream class, there are existed stream objects 39a to 39c corresponding to multi-media data 40a to 40c.

[0048]

For example, the contents object 38 represents such event as "news at 7:00 p.m. on September 1, 1995" or "an athletic event in 1995." Every contents object 38 has the time at which the event occurs, the length of time of the event, a keyword representing the event and the like as attributes thereof. The contents object 38 has a pointer to the plurality of stream objects 39a to 39c representing the contents object 38.

[0049]

The contents object 38 is logical representation of the event, while the stream objects 39a to 39c are physical representation of the event. The stream objects 39a to 39c correspond to the digital stream data of the event represented by the contents. In the case where an internal managing mechanism managing the stream data itself in the data base is used as a storage system of the stream data, the stream objects 39a to

39c have digital media data objects as attributes thereof, and in the case where the external managing mechanism is used as the storage system of the stream data, the stream data itself is managed in a file system of the OS 27, and a pointer thereof is managed in the data base.

[0050]

There is a plurality of methods of converting the stream data into digital data, so that several kinds of data different from one another can be prepared, in consideration of the environment in which it is used, the load required for reproduction, the quality thereof and the like, and relate them to the contents objects 38 as the stream objects 39a to 39c.

[0051]

For example, the contents object representing moving images of the above-mentioned "news at 7:00 p.m. on September 1, 1995" is recorded by using the following coding methods:

- MPEG-1video, frame size : 320 x 240, picture rate : 30fps,
- MPEG-1video, frame size : 320 x 240, picture rate : 15fps,
- MPEG-1video, frame size : 160 x 112, picture rate : 30fps,
- Motion-JPEG, frame size : 320 x 240, picture rate : 15fps,

and the stream objects corresponding to each of them are prepared and related to the contents objects.

[0052]

The stream objects has at least the following attributes:

- a pointer to the substance of data (in the case of internal management, a media data object, and in the case of external management, a file name),
 - information concerning the load required for reproducing the media data appropriately by a specific system (WS), and the name of the system,
 - a coding rate, in the case where the data is sound.
-

[0086]

Next, a view function of the multi-media retrieving system of one embodiment of the present invention will be described. A view is an object to be used for accessing to a part of the contents, and if the contents are the event concerning time series data such as the moving images and sound, a position of the view to the contents can be indicated by using a time taken from the beginning of the contents and a length of time for the view.

The time taken from the beginning of the contents and the length of time for the view are recorded as attributes of a view object. Further, the view function is realized in the contents class 41.

[0087]

For example, as shown in Fig. 12, a view B can be set in the contents A by designating skipLength which is the time taken for skipping from the beginning of the contents A, and timeLength which is the length of time for the view.

[0088]

When a view object is actually presented to the user, a part of the view in the contents can be recognized by the contents corresponding to the view, and by the skipLength which is the time taken from the beginning of the contents and the timeLength which is the length of time for the view. Thus, the view function can be provided to the user, by using the reproduction program 19 capable of reproducing the designated part of the contents.

[0089]

Fig. 13 is a diagram showing a relation between the view object, the contents object and the stream objects. In Fig. 13, an objectA is the contents object, an objectB is the view object, and an objectS1 and an objectS2 are the stream objects.

[0090]

It is assumed here that the timeLength is the length of time for the contents of the contents object A, which is, for example, 3600 seconds, keywords are collected keywords to be used for retrieving the contents, namely "news", streams S1 and S2 are collected sets of the stream objects representing the contents, there is not existed a parent which is the original contents in the case where the contents have been partially extracted by the information view, a child which is the partial contents corresponding to the contents, is B, the skipLength which is the time taken for skipping from the beginning of the contents is 0 second, and absStartTime which is the time at which the event occurs is 19:00:00 on September 1, 1995.

[0091]

In this case, the length of reproduction time of the contents object A is 3600 seconds, and it has the objectS1 and the objectS2 as the stream objects for displaying the contents, and the objectB as the view object.

[0092]

It is further assumed here that the timeLength is the length of time for the view of the view object B, which is, for example, 60 seconds, the keywords are the collected keywords to be used for retrieving the view, namely "headline," the streams S1 and S2 are the collected sets of stream objects representing the view, the parentA is the original contents of the partial contents extracted by the view, there is not existed the child which is the partial contents corresponding to the view, the skipLength is the time taken for skipping from the beginning of the contents which is 180 seconds, and the absStartTime which is the time at which the event occurs is 19:03:00 on September 1, 1995.

[0093]

In this case, the view object B corresponds to a period of 60 seconds after 180 seconds from the beginning of the contents object A. It has the objectS1 and the objectS2 as the stream objects for displaying the view.

[0094]

It is still further assumed here that contents A is the contents to which the stream objects S1 corresponds, a filename /usr/video/30fps.mpeg is a file name of actual multi-media data corresponding to the stream object S1, a load 35 is the load in displaying the multi-media data, a frameRate 30 is a frame rate, there is not existed a startFrameNo which is a discrepancy in agreement with the starting point in the contents class, a width 160 is a width of the frame, and a height 112 is a height of the frame.

[0095]

In this case, the stream object S1 corresponds to the actual multi-media data of the contents object A, the frame size is 160 x 112 PIXEL, and the picture rate is 30fps.

[0096]

It is furthermore assumed here that the contents A corresponds to the stream object S2, the fileName /user/video/10fps.mpeg is the filename of the actual multi-media data corresponding to the stream object S2, the load 15 is the load in displaying the multi-media data, the frameRate 10 is the frame rate, there is not existed the startFrameNo which is the discrepancy in the agreement with the starting point of the contents class, the width 160 is the width of the frame, and the height 112 is the height of the frame.

[0097]

In this case, the stream object S2 corresponds to the actual multi-media data of the contents object A, the frame size is 160 x 112 PIXEL, and the picture rate is 10fps.

[0098]

Fig. 14 is a flow chart showing reproducing processes of the view object. First, as shown in Step S1, the view object and a display start time of the view object are inputted.

[0099]

Next, as shown in Step S2, the contents object corresponding to the view object is obtained from the parent which is an attribute of the view object. And next, as shown in Step S3, the appropriate stream object is selected from the streams which are attributes of the contents object.

[0100]

Next, as shown in Step S4, the stream data corresponding to the selected stream object is reproduced for the period of the timeLength which is the length of time for the view after the skipLength which is an attribute of the view object.

[0101]

As described above, in order to realize the view, the given continuous media data should be able to be reproduced from the designated point. A reproduction start point of the multi-media data can be designated, by using the number of frames counted from the beginning and the like in the case of the moving images, and by using the reproducing time from the beginning and the like in the case of sound.

[0102]

For example, in the MPEG-1 video format, a unit named GOP (Group of Pictures) is used, wherein the data, in which several frames are coded, are collected into one group.

[0103]

Then, in order to reproduce the MPEG-1 video data 36 from any frame, there is a method of scanning the data from the beginning to the designated frame while analyzing header information, and then starting to display it.

[0104]

However, this method is inefficient because the data is scanned from the beginning even if the data is enormous. In the meantime, if it is possible to recognize in which GOP the designated frame is included, and in which number (offset) the designated frame is located among the frames within the GOP,

the reproduction can be started.

[0105]

Therefore, in the case of the MPEG-1video data 36, as an attribute of the stream object, the corresponding relationship between a start byte of a GOP header within the data and a frame number to be reproduced first within the GOP is checked in advance, and kept in the form of a corresponding table. Then, in the case of reproducing the data from a certain frame, the highest frame number not exceeding the designated frame number are obtained from the corresponding table, a difference between the highest frame number and the designated frame number is set to be the offset, and the start byte of the GOP is obtained.

[0106]

In the case of the MPEG-1audio data 37 of a fixed coding rate, the coding rate of the data is kept as the attribute of the stream object. Then, in the case of accessing to the data in the middle thereof, an access start point of the data is calculated by using the length of time taken for skipping and the coding rate.

[0107]

For example, the MPEG-1audio data 36 is composed of a series of packets in which the data is coded, and the MPEG-1audio data 37 has a mode composed of packets having almost the same length thereto. Then, in the case where the coding rate of the data is Rbps (bit per second) and the number of seconds S is the time taken for skipping the reproduction, the nearest header to the position of the RS/8byte is retrieved from the beginning, and the reproduction is started from the packet having the retrieved header.

[0218]

Fig. 27 is a flow chart showing a method of continuously displaying the moving images on the same window of one embodiment of the present invention. First, as shown in Step S71, one or more <file name of the moving image data, reproduction start time, point of the data to which partial access is made, picture rate and length of time for the reproduction> are inputted.

[0219]

Next, as shown in Step S72, one window is generated on the window system 35. And next, as shown in Step S73, the data

to be displayed are lined up in the order of reproduction start time.

[0220]

Then, as shown in Step S74, each data is initialized to be reproduced from the point of the data to which partial access is made. Next, as shown in Step S75, no process is made until the reproduction start time of the data to be reproduced first.

[0221]

Next, as shown in Step S76, the data reaching the reproduction start time is reproduced on the window. And next, as shown in Step S77, it is determined whether or not there is still more data to be reproduced. If there is no more data to be reproduced, the processes are completed, and if there is still more data to be reproduced, Step S77 proceeds to Step S78, advances the process to the next data and returns to the Step S76.

[0222]

Fig. 28 is a flow chart showing a method of continuously displaying the MPEG-1 video data 36 on the same window of one embodiment of the present invention. First, as shown in Step S81, one or more <file name of the MPEG-1 video data 36, reproduction start time, GOP to which the partial access is made, picture rate and length of reproduction time> are inputted.

[0223]

Next, as shown in Step S82, one window is generated on the window system 35. Then, as shown in Step S83, the data to be reproduced are lined up in the order of reproduction start time.

[0224]

Next, as shown in Step S84, each data is decoded without displaying the frame for the offset upon analyzing the sequence header and moving the file pointer to the designated GOP.

[0025]

And next, as shown in Step S85, no process is made until the reproduction start time of the data to be reproduced first. Then, as shown in Step S86, the data reaching the reproduction start time is reproduced on the window.

[0226]

Next, as shown in Step S87, it is determined whether or not there is still more data to be reproduced. If there is no more data to be reproduced, the processes are completed, and if there is still more data to be reproduced, the Step S87 proceeds

to Step S88, advances the process to the next data and returned to the Step S86.

[0227]

Fig. 29 is a flow chart showing a method of displaying the view of one embodiment of the present invention by using the plurality of stream data. First, as shown in Step S91, <the file names of the moving images, the reproduction start times, the points of the data to which partial accesses are made, the picture rates and the lengths of the reproduction times> are inputted to the view by using the scheduler 18.

[0228]

Next, as shown in Step S92, one window is generated on the window system 35. Then, as shown in Step S93, the data to be displayed are lined up in the order of reproduction start time.

[0229]

And next, as shown in Step S94, initialization is performed to each data, for the purpose of the reproduction from the point of the data to which the partial access is made. Next, as shown in Step S95, no process is made until the reproduction start time of the data to be reproduced first.

[0230]

Then, as shown in Step S96, the data reaching the reproduction start time is reproduced on the window. Next, as shown in Step S97, it is determined whether or not there is still more data to be reproduced. If there is no more data to be reproduced, the processes are completed, and if there is still more data to be reproduced, Step S97 proceeds to Step S98, advances the process to the next data and returned to Step S96.

(19) 日本国特許庁 (JP)

公開特許公報 (A)

(11) 特許出願公開番号

特開平 9 - 8 1 4 4 4

(43) 公開日 平成9年(1997)3月28日

(51) Int. Cl. ⁸	識別記号	庁内整理番号	FI	技術表示箇所
G 0 6 F	12/00	5 4 7	G 0 6 F	12/00 5 4 7 D
		5 2 0		5 2 0 A
	17/30		H 0 4 N	5/78 5 1 0 Z
H 0 4 N	5/78	5 1 0		7/173
	7/173		G 0 6 F	15/40 3 7 0 G
審査請求 未請求 請求項の数 2 3			OL	(全 4 2 頁)

(21) 出願番号 特願平7-240351

(22) 出願日 平成7年(1995)9月19日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番1号

(72) 発明者 加藤 光幾

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72) 発明者 近藤 朗子

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72) 発明者 石川 博

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74) 代理人 弁理士 大菅 義之 (外1名)

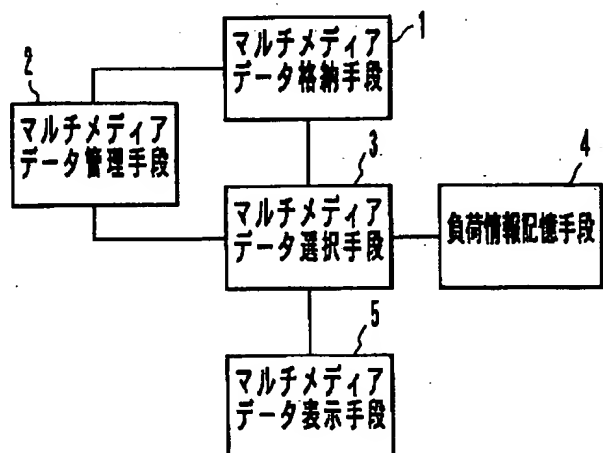
(54) 【発明の名称】 マルチメディアデータ検索システム、マルチメディアデータのスケジューリング装置、マルチメディアデータの再生方法及び再生プログラム記憶媒体

(57) 【要約】

【課題】 マルチメディアデータを再生するために必要な負荷を制御することができるマルチメディアデータ検索システムを提供する。

【解決手段】 マルチメディアデータ管理手段2は、事象に関する事柄を記述するコンテンツの管理と、コンテンツに対応しコンテンツの実体を表すマルチメディアデータの管理とを独立に行い、1つのコンテンツに対してマルチメディアデータ格納手段1に格納された複数のマルチメディアデータが対応できるようにし、検索したマルチメディアデータをユーザへの提示する場合、マルチメディアデータ表示手段5に表示する際の負荷を負荷情報記憶手段4から読み出し、複数のマルチメディアデータの中から提示に用いるマルチメディアデータをマルチメディアデータ選択手段3により選択する。

本発明の一実施例によるマルチメディアデータ検索システムの機能的な構成を示すブロック図



【特許請求の範囲】

【請求項1】 1つの事象に対応する複数のマルチメディアデータを生成するマルチメディアデータ生成手段と、

前記事象に対応する複数のマルチメディアデータの中から1つを選択するマルチメディアデータ選択手段とを備えることを特徴とするマルチメディアデータ検索システム。

【請求項2】 1つの事象に対応する複数のマルチメディアデータを格納するマルチメディアデータ格納手段と、

1つの事象に対応する複数のマルチメディアデータに対し、前記事象を表す共通の識別子に対応させるマルチメディアデータ管理手段と、

前記識別子を指定することにより、前記識別子に対応する複数のマルチメディアデータを選択し、この複数のマルチメディアデータの中から1つを選択するマルチメディアデータ選択手段とを備えることを特徴とするマルチメディアデータ検索システム。

【請求項3】 1つの事象に対応する複数のマルチメディアデータを格納するマルチメディアデータ格納手段と、

前記複数のマルチメディアデータに対し、前記事象を表す共通の識別子に対応させるマルチメディアデータ管理手段と、

前記マルチメディアデータを表示するマルチメディアデータ表示手段と、

前記マルチメディアデータを表示する際の処理に必要な負荷量を、前記複数のマルチメディアデータのそれぞれに対応させて記憶する負荷情報記憶手段と、

前記負荷量に基づいて、共通の識別子を有する複数のマルチメディアデータの中から1つを選択するマルチメディアデータ選択手段とを備えることを特徴とするマルチメディアデータ検索システム。

【請求項4】 連続メディアデータから品質の異なるデータを生成するマルチメディアデータ生成手段と、

前記マルチメディアデータ生成手段で生成したデータをコンテンツと対応付けるマルチメディアデータ管理手段と、

前記コンテンツを具象させる場合、前記コンテンツに対応づけられている1つ以上のデータの中から適切なものを選択するマルチメディアデータ選択手段とを備えることを特徴とするマルチメディアデータ検索システム。

【請求項5】 前記マルチメディアデータ生成手段は、ピクチャレートの r フレーム/秒の動画データに対し、フレームを n 個置きに選択することにより、ピクチャレートの $r/(n+1)$ フレーム/秒の動画データを生成することを特徴とする請求項4に記載のマルチメディアデータ検索システム。

【請求項6】 前記マルチメディアデータ生成手段は、

画像データのフレームサイズを小さくしたフレームを生成し、前記フレームによりフレームサイズの小さい画像データを生成することを特徴とする請求項4に記載のマルチメディアデータ検索システム。

【請求項7】 マルチメディアデータの内容の一部を抽出してビューを設定するビュー生成手段と、

1つ以上のビューに対応するコンテンツとそれぞれのビューの表示開始時刻情報とに基づいて、前記コンテンツに対応するマルチメディアデータから適当なものを選択し、前記表示開始時刻から表示するように管理するスケジューリング手段とをさらに備えることを特徴とする請求項1～6のいずれか1項に記載のマルチメディアデータ検索システム。

【請求項8】 マルチメディアデータの内容の一部のビューの再生開始時刻と再生時間長との組の情報から、前記ビューの再生が重なる時間帯を算出する時間帯算出手段と、前記各時間帯ごとに、前記ビューに対応するコンテンツを検索するコンテンツ検索手段と、

前記コンテンツに対応するマルチメディアデータの再生処理に必要な負荷量を、前記マルチメディアデータの品質属性から算出する負荷量算出手段と、

前記負荷量に基づいて、再生に使用するマルチメディアデータを選択するマルチメディアデータ選択手段とを備えることを特徴とするマルチメディアデータのスケジューリング装置。

【請求項9】 前記負荷量算出手段は、

MPEGデータのIフレームを指定レートで表示するのに必要な単位時間当たりの処理時間を、前記負荷量として用いることを特徴とする請求項8に記載のマルチメディアデータのスケジューリング装置。

【請求項10】 前記マルチメディアデータ選択手段は、

前記コンテンツにMPEGデータが対応する場合、各時間帯ごとの前記負荷量が100%を越えないように、再生に使用するマルチメディアデータを選択することを特徴とする請求項8に記載のマルチメディアデータのスケジューリング装置。

【請求項11】 前記マルチメディアデータ選択手段は、

前記コンテンツに対応する動画データにおいて、フレームサイズとピクチャレートとが異なる画像が存在する場合、フレームサイズが大きい方を優先して選択し、フレームサイズが同一でピクチャレートが異なる場合、ピクチャレートの大きい方を優先して選択することを特徴とする請求項8に記載のマルチメディアデータのスケジューリング装置。

【請求項12】 前記マルチメディアデータ選択手段は、

ビューの再生時において、同一のビューに対して同一サイズのフレームを採用することを特徴とする請求項8に

記載のマルチメディアデータのスケジューリング装置。

【請求項13】 前記マルチメディアデータ選択手段は、

特定の時間帯におけるビューの再生に必要な負荷量が所定の値を越える場合、前記特定の時間帯におけるビューの再生に用いるフレームサイズを小さいサイズに変更し、且つ他の時間帯で用いられている前記ビューのフレームサイズを前記特定の時間帯における前記ビューのフレームサイズに一致させることを特徴とする請求項12に記載のマルチメディアデータのスケジューリング装置。

【請求項14】 前記マルチメディアデータ選択手段は、

ビューに対応する各ストリームデータに優先順位を付与し、

特定の時間帯におけるビューの再生に必要な負荷量の合計が所定の値を越える場合、前記特定の時間帯におけるビューの再生に用いるストリームデータを負荷量の低い優先順位のものに変更し、

前記ビューの再生に用いるストリームデータの優先順位を変更した結果、前記ビューの再生に用いるフレームサイズが変化した場合、前記ビューの全ての時間帯における優先順位を、前記特定の時間帯における優先順位に一致させ、

前記特定の時間帯より前の各ビューの再生に用いるストリームデータを、同一フレームサイズの範囲内で最も優先度の高いものに変更し、

最初の時間帯からストリームデータの選択をやり直すことを特徴とする請求項8に記載のマルチメディアデータのスケジューリング装置。

【請求項15】 前記マルチメディアデータ選択手段は、

前記ビューが動画像の場合、前記ビューを再生する時間帯を数分割し、マルチメディアデータのファイル名、再生開始時刻、部分アクセスする場所、再生時間長の組を各時間帯ごとに指定して再生時のピクチャレートを変化させることを特徴とする請求項8に記載のマルチメディアデータのスケジューリング装置。

【請求項16】 前記コンテンツの表示場所が指定されていない場合、前記マルチメディアデータを表示するウィンドウ同士が重ならないように表示場所を決定するウィンドウ設定手段をさらに備えることを特徴とする請求項8に記載のマルチメディアデータのスケジューリング装置。

【請求項17】 MPEG-1 videoデータにおけるGOPヘッダの開始バイトと、GOP内で最初に再生されるフレーム番号との対応表を、前記MPEG-1 videoデータの属性として備え、

前記MPEG-1 videoデータを指定フレームから再生する場合、前記対応表を用いることにより、前記指

定フレームが含まれるGOPとGOPの先頭から何フレーム目であるかを示すオフセットとを前記指定フレームの番号から計算し、

シーケンスヘッダを解析した後、得られたGOPまでファイルポインタを移動し、

前記オフセット分のフレームを表示することなく、前記MPEG-1 videoデータを復号し、

前記指定フレームから前記MPEG-1 videoデータの再生を開始することを特徴とするマルチメディアデータの再生方法。

【請求項18】 MPEG-1 audioデータにおける符号化レートを、前記MPEG-1 audioデータの属性として取得し、

前記MPEG-1 audioデータの途中からアクセスする場合、スキップする時間長と前記符号化レートとに基づいて、前記MPEG-1 audioデータのアクセス開始箇所を計算し、

前記アクセス開始箇所に最も近いフレームヘッダを検索し、

前記MPEG-1 audioデータの再生を前記フレームヘッダから開始することを特徴とするマルチメディアデータの再生方法。

【請求項19】 同一サイズのフレームを持つ複数のマルチメディアデータに対し、各マルチメディアデータのファイル名、各マルチメディアデータの再生開始時刻、各マルチメディアデータの部分アクセスをする場所、各マルチメディアデータのピクチャレート、各マルチメディアデータの再生時間長を指定し、

各マルチメディアデータの再生される時間帯が重なら

ず、且つ連続している場合、前記各マルチメディアデータを前記再生開始時刻よりウィンドウ上に表示を開始し、

前記部分アクセスをする場所から前記ピクチャレートで前記再生時間長の間、前記各マルチメディアデータを再生することを特徴とするマルチメディアデータの再生方法。

【請求項20】 最初のMPEG-1 videoデータの再生開始時刻以前に表示用ウィンドウを生成し、

各MPEG-1 videoデータのシーケンスヘッダ部分を読み込み初期化を行い、

指定されたGOPバイト及びオフセットから表示する状態で再生開始時刻を待ち、

各MPEG-1 videoデータを前記再生開始時刻から連続して再生することを特徴とするマルチメディアデータの再生方法。

【請求項21】 ピクチャレートがrフレーム/秒のMPEG-1 videoデータのフレームSから表示を行う場合、

フレームnのピクチャスタートコードを復号した時点の時刻T₁を取得し、表示開始時刻T₀からの時間が

10

20

30

40

50

5

$(T_1 - T_0) > (n - S) / r$ で、且つフレーム n が B フレームの場合、フレーム n の復号をせず、フレーム n の復号が終了した時点の時刻 T_2 を取得し、表示開始時刻 T_0 からの時間が $(T_2 - T_0) > (n - S) / r$ の場合、表示を行わないようにすることを特徴とするマルチメディアデータの再生方法。

【請求項22】 マルチメディアデータを表示する際の処理に必要な負荷量に基づいて、事象を表す共通の識別子を与えられた複数のマルチメディアデータの中から1つを選択して表示するプログラムを記憶したことを特徴とする再生プログラム記憶媒体。

【請求項23】 符号化された複数のマルチメディアデータを予め復号しておき、各マルチメディアデータの表示開始時刻になった時点で、復号したマルチメディアデータを順次に表示するプログラムを記憶したことを特徴とする再生プログラム記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、マルチメディアデータ検索システム、マルチメディアデータのスケジューリング装置、マルチメディアデータの再生方法及び再生プログラム記憶媒体に関し、マルチメディアデータベースのデータ管理やデータ再生の管理に適用して好適なものである。

【0002】

【従来の技術】 近年、マルチメディアの利用が盛んになっており、マルチメディアの蓄積、検索、再生を統合して行うプラットフォームとして、マルチメディアデータベース (MMDB) が有望である。

【0003】 マルチメディアデータベースは、テキスト、グラフィックス、動画像、音声など性質が異なるメディアに対し、データの符号化などの各メディアに特有な操作をユーザから隠蔽することにより、メディアを統合的に扱えるようにしたものである。

【0004】 マルチメディアデータの大きな特徴の1つは、動画像、音声などの時系列データを含んでいることである。これらの時間に関して連続した連続メディアデータは、適切な再生スピード (例えば、動画像を記録した時と同じピクチャレート) で再生する必要がある。

【0005】 また、動画像と音声とを同時に出力したりする場合などにおいて、メディア間の同期を取るために、指定時刻から正確に再生を開始し、再生中に処理の遅延が許されないという要求がある。

【0006】 さらに、マルチメディアデータを検索した結果として、複数の動画像を同時に表示する場合、データの復号にコンピュータの資源を大量に用いるため、資源の適切な配分を行う必要がある。

【0007】 一方、連続メディアはそのデータ量が膨大である。そして、マルチメディアデータは、その内容によって意味を持つ時間帯に分割され、分割したそれぞれ

6

の時間帯が重なりを有する場合がある。

【0008】 例えば、動画像に映っている物体は、それぞれある時刻に出現し、所定の時間が経過したのち消滅する。画面上には、複数の物体が同時に映っている場合があり、注目する物体ごとに、連続メディアの時間帯の範囲が異なる。そのため、連続メディアデータを予め細かく分割して管理するのは好ましくなく、連続メディアデータを連続したまま扱い、必要に応じて連続メディアデータの一部分にアクセスできるようにするべきである。

【0009】 そのため、マルチメディアデータベースでは、マルチメディアデータの一部分を検索できるようなデータモデルと、検索した時間帯をユーザに提示する場合に、連続メディアを連続したまま扱い、且つマルチメディアデータの一部分にアクセスできる方法が必要となっていた。

【0010】

【発明が解決しようとする課題】 従来のマルチメディアデータの格納方法は、マルチメディアデータとそのマルチメディアデータが記録している事象とを1対1に対応させて格納するようにしていた。このため、例えば、ある事象を品質の異なる複数の方法でマルチメディアデータに記録しても、その事象が異なる品質のマルチメディアデータで記録されていることを表現する方法がなかった。

【0011】 また、マルチメディアデータの品質を選択する機能が提供されないため、資源に余裕がある場合や資源が不足している場合でも、1つのマルチメディアデータを用いなければならず、コンピュータ資源に対応してマルチメディアデータの品質を変えることができなかった。

【0012】 また、マルチメディアデータ検索システムは、マルチメディアデータをひとまとまりとしてとらえているため、マルチメディアデータの一部分にアクセスすることができないという問題があった。

【0013】 また、マルチメディアデータをユーザに提示するためのマルチメディアデータの復号には、ハードウェアを用いる方法とソフトウェアを用いる方法とがある。ハードウェアを用いる復号では、複数のストリームを同時に復号することができないため、動画像を複数個同時に表示することができないという問題があった。。

【0014】 また、ソフトウェアを用いる復号では、複数のストリームを同時に復号することはできるが、コンピュータの資源に限りがあるため、動画像の場合、再生時間内に正しく再生しようとするフレームを飛ばして表示を行わなければならない、フレームが不定の間隔で表示され見苦しい画像となり、無駄な処理が生じるという問題があった。また、フレームを飛ばさないように再生しようとした場合、正しい時間で表示できず、再生時間の遅れが生じるため、他のマルチメディアデータとの同

期を保つことができないという問題があった。

【0015】また、スクリプト（データの検索、表示などを指示する記述）を実行する場合など、メディアが再生される時間帯が予め分かっており、資源量が途中で変化することが事前に予測できても、それに対応でない。

【0016】そこで、本発明の目的は、マルチメディアデータを再生するために必要な負荷を制御することができる、マルチメディアデータの一部分を検索することができるマルチメディアデータ検索システム、マルチメディアデータのスケジューリング装置、マルチメディアデータの再生方法及び再生プログラム記憶媒体を提供することである。

【0017】

【課題を解決するための手段】 上述した課題を解決するために、請求項1の発明によれば、1つの事象に対応する複数のマルチメディアデータを生成し、1つの事象に対応する複数のマルチメディアデータの中から1つを選択する。このことにより、マルチメディアデータを事象ごとに管理することができ、1つの事象に複数のマルチメディアデータを対応させることができるため、1つの事象に対応する異なる品質のマルチメディアデータを用意しておくことにより、マルチメディアデータを再生するために必要な負荷を制御することができる。ここで、事象とは、例えば「7時ニュース」のときは、一般的意味における7時のニュースのことであり、後述するコンテンツ（内容）とはこの事象とほぼ同義であるがこの事象をコンピュータで扱うための表現である。

【0018】また、請求項2の発明によれば、1つの事象に対応する複数のマルチメディアデータに対し、前記事象を表す共通の識別子に対応させ、識別子を指定することにより、識別子に対応する複数のマルチメディアデータの中から1つを選択する。このことにより、1つの事象に対応する複数のマルチメディアデータを容易に取り出すことができる。

【0019】また、請求項3の発明によれば、複数のマルチメディアデータに対し、事象を表す共通の識別子に対応させ、マルチメディアデータを表示する際の処理に必要な負荷量を、複数のマルチメディアデータのそれぞれに対応させて記憶し、前記負荷量に基づいて、共通の識別子を有する複数のマルチメディアデータの中から1つを選択する。このことにより、マルチメディアデータを再生するために必要な負荷を制御しながら、1つの事象に対応する複数のマルチメディアデータから適切なものを容易に取り出すことができる。例えば、1つのコンテンツに対応して異なった品質を有する複数のマルチメディアデータから負荷に応じて適切なマルチメディアデータを選択することにより、フレーム飛ばしや再生遅れがなくマルチメディアデータを再生することができる。

【0020】また、請求項4の発明によれば、連続メディアデータから品質の異なるデータを生成し、生成した

データをコンテンツと対応付け、前記コンテンツを具象させる場合、前記コンテンツに対応づけられている1つ以上のデータの中から適切なものを選択する。このことにより、同一のコンテンツで異なる品質のマルチメディアデータから適切なものを選択することができ、マルチメディアデータを再生するために必要な負荷を制御することができる。

【0021】また、請求項5の発明によれば、ピクチャーレートが r フレーム/秒の動画データに対し、フレームを n 個置きに選択することにより、ピクチャーレートが $r/(n+1)$ フレーム/秒の動画データを生成する。このことにより、同一のコンテンツで異なる品質のマルチメディアデータを容易に生成することができる。

【0022】また、請求項6の発明によれば、画像データのフレームサイズを小さくしたフレームを生成し、前記フレームによりフレームサイズの小さい画像データを生成する。このことにより、同一のコンテンツで異なる品質のマルチメディアデータを容易に生成することができる。

【0023】また、請求項7の発明によれば、マルチメディアデータの内容の一部を抽出してビューを設定し、1つ以上のビューに対応するコンテンツとそれぞれのビューの表示開始時刻情報とに基づいて、前記コンテンツに対応するマルチメディアデータから適当なものを選択し、前記表示開始時刻から表示する。このことにより、マルチメディアデータの一部分を容易に検索することができる。

【0024】また、請求項8の発明によれば、マルチメディアデータの内容の一部のビューの再生開始時刻と再生時間長との組の情報から、前記ビューの再生が重なる時間帯を算出し、前記各時間帯ごとに、前記ビューに対応するコンテンツを検索し、前記コンテンツに対応するマルチメディアデータの再生処理に必要な負荷量を、前記マルチメディアデータの品質属性から算出し、前記負荷量に基づいて、再生に使用するマルチメディアデータを選択する。このことにより、同一のコンテンツで異なる品質のマルチメディアデータの一部分から適切なものを選択することができ、マルチメディアデータの一部分を再生するために必要な負荷を制御することができる。

【0025】また、請求項9の発明によれば、MPEGデータのIフレームを指定レートで表示するのに必要な単位時間当たりの処理時間を、負荷量として用いる。このことにより、マルチメディアデータを再生するために必要な負荷量を短時間で容易に求めることができる。

【0026】また、請求項10の発明によれば、コンテンツにMPEGデータが対応する場合、各時間帯ごとの前記負荷量が100%を越えないように、再生に使用するマルチメディアデータを選択する。このことにより、マルチメディアデータの再生に用いる資源の範囲内でマ

ルチメディアデータの再生を行うことができる。

【0027】また、請求項11の発明によれば、コンテンツに対応する動画データにおいて、フレームサイズとピクチャレートとが異なる画像が存在する場合、フレームサイズが大きい方を優先して選択し、フレームサイズが同一でピクチャレートが異なる場合、ピクチャレートの大きい方を優先して選択する。このことにより、マルチメディアデータの再生に用いる資源の範囲内で最適な品質のマルチメディアデータの再生を行うことができる。

【0028】また、請求項12の発明によれば、ビューの再生時において、同一のビューに対して同一サイズのフレームを採用する。このことにより、再生の途中にフレームサイズが変化して、画面が見にくくなることを回避できる。

【0029】また、請求項13の発明によれば、特定の時間帯におけるビューの再生に必要な負荷量が所定の値を越える場合、前記特定の時間帯におけるビューの再生に用いるフレームサイズを小さいサイズに変更し、且つ他の時間帯で用いられている前記ビューのフレームサイズを前記特定の時間帯における前記ビューのフレームサイズに一致させる。このことにより、再生の途中にフレームサイズが変化して、画面が見にくくなることを防止できる。

【0030】また、請求項14の発明によれば、ビューに対応する各ストリームデータに優先順位を付与し、特定の時間帯におけるビューの再生に必要な負荷量の合計が所定の値を越える場合、前記特定の時間帯におけるビューの再生に用いるストリームデータを負荷量の低い優先順位のものに変更し、前記ビューの再生に用いるストリームデータの優先順位を変更した結果、前記ビューの再生に用いるフレームサイズが変化した場合、前記ビューの全ての時間帯における優先順位を、前記特定の時間帯における優先順位に一致させ、前記特定の時間帯より前の各ビューの再生に用いるストリームデータを、同一フレームサイズの範囲内で最も優先度の高いものに変更し、最初の時間帯からストリームデータの選択をやり直す。このことにより、複数のビューを同時に表示する場合において、再生の途中にフレームサイズが変化して、画面が見にくくなることを防止しながら、マルチメディアデータの再生に用いる負荷量を資源の範囲内に抑えることができる。

【0031】また、請求項15の発明によれば、ビューが動画の場合、ビューを再生する時間帯を数分割し、マルチメディアデータのファイル名、再生開始時刻、部分アクセスする場所、再生時間長の組を各時間帯ごとに指定して再生時のピクチャレートを変化させる。このことにより、マルチメディアデータの一部分を再生するために必要な負荷を制御することができる。

【0032】また、請求項16の発明によれば、コンテ

ントの表示場所が指定されていない場合、マルチメディアデータを表示するウィンドウ同士が重ならないように表示場所を決定する。このことにより、複数のビューが重なったまま同時に再生されることを防止できる。

【0033】また、請求項17の発明によれば、MPEG-1 videoデータにおけるGOPヘッダの開始バイトと、GOP内で最初に再生されるフレーム番号との対応表を、前記MPEG-1 videoデータの属性として備え、MPEG-1 videoデータを指定フレームから再生する場合、前記対応表を用いることにより、前記指定フレームが含まれるGOPとGOPの先頭から何フレーム目であるかを示すオフセットとを前記指定フレームの番号から計算し、シーケンスヘッダを解析した後、得られたGOPまでファイルポインタを移動し、前記オフセット分のフレームを表示することなく、前記MPEG-1 videoデータを復号し、前記指定フレームから前記MPEG-1 videoデータの再生を開始する。このことにより、MPEG-1 videoデータを指定フレームから正確に再生することができる。

【0034】また、請求項18の発明によれば、MPEG-1 audioデータにおける符号化レートを、前記MPEG-1 audioデータの属性として取得し、前記MPEG-1 audioデータの途中からアクセスする場合、スキップする時間長と前記符号化レートとに基づいて、前記MPEG-1 audioデータのアクセス開始箇所を計算し、前記アクセス開始箇所に最も近いフレームヘッダを検索し、前記MPEG-1 audioデータの再生を前記フレームヘッダから開始する。このことにより、MPEG-1 audioデータを途中から正確に再生することができる。

【0035】また、請求項19の発明によれば、同一サイズのフレームを持つ複数のマルチメディアデータに対し、各マルチメディアデータのファイル名、各マルチメディアデータの再生開始時刻、各マルチメディアデータの部分アクセスをする場所、各マルチメディアデータのピクチャレート、各マルチメディアデータの再生時間長を指定し、各マルチメディアデータの再生される時間帯が重ならず、且つ連続している場合、前記各マルチメディアデータを前記再生開始時刻よりウィンドウ上に表示を開始し、前記部分アクセスをする場所から前記ピクチャレートで前記再生時間長の間、前記各マルチメディアデータを再生する。このことにより、複数のマルチメディアデータを指定した再生開始時刻から連続して再生することができる。

【0036】また、請求項20の発明によれば、最初のMPEG-1 videoデータの再生開始時刻以前に表示用ウィンドウを生成し、各MPEG-1 videoデータのシーケンスヘッダ部分を読み込み初期化を行い、指定されたGOPバイト及びオフセットから表示する状態で再生開始時刻を待ち、各MPEG-1 videoデ

10

20

30

40

50

ータを前記再生開始時刻から連続して再生する。このことにより、各MPEG-1 videoデータを指定した再生開始時刻になった時点で即座に再生することができ、複数のMPEG-1 videoデータを連続して再生することができる。

【0037】また、請求項21の発明によれば、ピクチャレートが r フレーム/秒(fps)のMPEG-1 videoデータのフレーム S から表示を行う場合、フレーム n のピクチャスタートコードを復号した時点の時刻 T_1 を取得し、表示開始時刻 T_0 からの時間が $(T_1 - T_0) > (n - S) / r$ で、且つフレーム n が B フレームの場合、フレーム n の復号をせず、フレーム n の復号が終了した時点の時刻 T_2 を取得し、表示開始時刻 T_0 からの時間が $(T_2 - T_0) > (n - S) / r$ の場合、表示を行わないようにする。このことにより、表示開始時刻までに復号や表示が間に合わないMPEG-1 videoデータの復号や表示をスキップすることができ、資源を有効に使うことができる。

【0038】また、請求項22の発明によれば、マルチメディアデータを表示する際の処理に必要な負荷量に基づいて、事象を表す共通の識別子を与えられた複数のマルチメディアデータの中から1つを選択して表示するプログラムを記憶しておく。このことにより、同一のコンテンツの複数のマルチメディアデータから適切なものを選択することができ、マルチメディアデータを再生するために必要な負荷を制御することができる。

【0039】また、請求項23の発明によれば、符号化された複数のマルチメディアデータを予め復号しておき、各マルチメディアデータの表示開始時刻になった時点で、復号したマルチメディアデータを順次に表示するプログラムを記憶しておく。このことにより、復号に要する時間に影響されることなく、複数のマルチメディアデータを連続して表示することができる。

【0040】

【発明の実施の形態】以下、本発明の一実施例によるマルチメディアデータ検索システムについて、図面を参照しながら説明する。

【0041】図1において、マルチメディアデータ格納手段1は、1つの事象に対応する複数のマルチメディアデータを格納するものである。また、マルチメディアデータ管理手段2は、複数のマルチメディアデータに対し、事象を表す共通の識別子に対応させるものである。また、マルチメディアデータ選択手段3は、表示に必要な負荷量に基づいて、共通の識別子を有する複数のマルチメディアデータの中から1つを選択するものである。また、負荷情報記憶手段4は、マルチメディアデータの情報を表示する際の処理に必要な負荷量を、複数のマルチメディアデータのそれぞれに対応させて記憶するものである。また、マルチメディアデータ表示手段5は、マルチメディアデータ情報を表示するものである。

【0042】そして、マルチメディアデータ管理手段2において、事象に関する事柄を記述するコンテンツの管理と、コンテンツに対応しコンテンツの実体を表すマルチメディアデータの管理とを独立に行うことにより、1つのコンテンツに対して複数のマルチメディアデータが対応できるようにする。また、検索したマルチメディアデータをユーザへの提示する場合、負荷情報記憶手段4に記憶された負荷量を参照することにより、コンピュータ資源を考慮しながら複数のマルチメディアデータの中から提示に用いるマルチメディアデータを選択する。

【0043】図2は、本発明の一実施例によるマルチメディアデータ検索システムの構成を示すブロック図である。図2において、11はCPUであり、データベースエンジン26及びOS27が動作する。12は記憶装置であり、データベースが管理するOBデータ13と、必要に応じてメディアデータ20が格納される。13はオブジェクトベース(OB)であり、クラス情報14とメソッド15とを記憶する。クラス情報14はコンテンツ16を管理するためのインスタンスとストリーム17を管理するためのインスタンスを持つ。

【0044】メソッド15はメディアデータ20を再生する際に使用するスケジューラ18、再生プログラム19、keyword()メソッド28、setView()メソッド29、newVideo()メソッド30、newAudio()メソッド31、newRate()メソッド32、newSize()メソッド33、play()メソッド34を含む。メディアデータ20はデータベースの外部管理機構を使う場合に用いるもので、マルチメディアデータを格納する。21はメモリである。22はI/Oであり、記憶装置12とデータ入力をするためのキーボード23データ出力をするためのCRT24、スピーカ25及びCPU11を接続する。

【0045】また、図3に示すように、CPU11でウィンドウシステム35を管理し、メディアデータ20として、MPEG-1 videoデータ36とMPEG-1 audioデータ37とを格納している。

【0046】なお、本実施例では、データベースとしてオブジェクト指向データベース(OODB)を用い、オブジェクト指向データベースに基づいて、マルチメディアを扱うための機能を拡張する。

【0047】マルチメディアデータベースはオブジェクト指向データベースを基礎としており、オブジェクトをクラス階層で管理する。クラスとして事象を代表するコンテンツクラスと、コンテンツを実際に実現するためのデータを管理するストリームクラスとを設ける。コンテンツクラスのインスタンスとして、図4に示すように、特定の事象を代表するコンテンツオブジェクト38が存在し、ストリームクラスのインスタンスとして、マルチメディアデータ40a~40cに対応するストリームオ

プロジェクト39a~39cが存在する。

【0048】例えば、コンテンツオブジェクト38は、“1995年9月1日午後7時のニュース”や“1995年の運動会”などの事象を代表する。各コンテンツオブジェクト38は、その事象が起こった時刻、事象の時間長、事象を表すキーワードなどを属性として持つ。コンテンツオブジェクト38は、コンテンツオブジェクト38を表現する複数のストリームオブジェクト39a~39cへのポインタを持つ。

【0049】コンテンツオブジェクト38が事象の論理的な表現であるのに対し、ストリームオブジェクト39a~39cは事象の物理的な表現である。コンテンツオブジェクト38を実際に表現するストリームオブジェクト39a~39cは、コンテンツが表す事象をデジタル化したストリームデータに対応する。ストリームデータの格納方式として、ストリームデータ自体をデータベースで管理する内部管理機構を用いる場合、デジタル化したメディアデータオブジェクトをストリームオブジェクト39a~39cの属性として持ち、外部管理機構を用いる場合、ストリームデータ自体をOS27のファイルシステムで管理し、データベースではそのポインタを管理する。

【0050】ストリームデータをデジタル化する方法は複数個存在するため、使用する環境、再生するために必要な負荷、品質等を考慮して数種類の異なるデータを作成し、ストリームオブジェクト39a~39cとしてコンテンツオブジェクト38と関連付けることができる。

【0051】例えば、上述した“1995年9月1日午後7時のニュース”の動画像を表すコンテンツオブジェクトに対し、

- ・MPEG-1 video、フレームサイズ：320×240、ピクチャレート：30fps

- ・MPEG-1 video、フレームサイズ：320×240、ピクチャレート：15fps

- ・MPEG-1 video、フレームサイズ：160×112、ピクチャレート：30fps

- ・Motion-JPEG、フレームサイズ：320×240、ピクチャレート：15fps

の符号化方法を用いて記録し、それぞれに対するストリームオブジェクトを作成し、コンテンツオブジェクトと関連付ける。

【0052】ストリームオブジェクトは、属性として少なくとも、

- ・データの実体へのポインタ（内部管理の場合、メディアデータオブジェクト、外部管理の場合、ファイル名）、

- ・メディアデータを特定の装置（WS）で適切に再生するために必要な負荷に関する情報とその装置の名前、

- ・データが動画像の場合、フレームサイズ、ピクチャレート、

- ・データが音声の場合、符号化レート、を持つ。

【0053】次に、オブジェクトを管理するクラス階層について説明する。図5において、コンテンツオブジェクトのクラス構造は、コンテンツオブジェクトの共通の属性を示すコンテンツクラス41の下位に、ビデオコンテンツクラス42、オーディオコンテンツクラス43及び図示しないイメージコンテンツクラス、テキストコンテンツクラスを有している。

【0054】コンテンツクラス41には属性として、コンテンツの時間長timeLength、コンテンツを検索するために用いるキーワードの集合を格納するkeywords、コンテンツを具象するストリームオブジェクトの集合を格納するstreams、コンテンツが情報ビューによって切り出された部分コンテンツである場合の元のコンテンツparent、コンテンツに対応する部分コンテンツchilds及びコンテンツが該当キーワードを有しているかどうかをチェックするkeyword()メソッドを有する。

【0055】コンテンツクラス41の下にはそれを継承するメディアごとのサブクラスが定義される。すなわち、ビデオコンテンツクラス42は、コンテンツの先頭からのスキップする時間skipLengthの属性、事象が生じた時刻absStartTimeの属性、コンテンツオブジェクトに対しビューオブジェクトを生成し部分コンテンツchildsに設定するsetView()メソッド、コンテンツオブジェクトに対しストリームオブジェクトを生成しストリームオブジェクトの集合streamsに格納するnewVideo()メソッド及びビューを再生するplay()メソッドを有する。

【0056】なお、事象が生じた時刻absStartTimeは、コンテンツオブジェクトの絶対時間を意味しており、この事象が生じた時刻absStartTimeを用いることにより、複数のコンテンツを同期させて表示することができる。

【0057】また、オーディオコンテンツクラス43は、コンテンツの先頭からの時間skipLengthの属性、事象の生じた時刻absStartTimeの属性、部分オーディオコンテンツを生成するsetView()メソッド、コンテンツに対応するオーディオストリームオブジェクトを生成するnewAudio()メソッド及びオーディオデータのコンテンツを表示するplay()メソッドを有する。

【0058】また、図6において、ストリームオブジェクトのクラス構造は、ストリームオブジェクトの共通の属性を示すストリームクラス51の下位に、ビデオクラス51、オーディオクラス53及び図示しないイメージクラス、テキストクラスを有している。さらに、各メディア毎のクラスの下位には、各メディア毎にフォーマッ

ト毎のクラスを有している。すなわち、ビデオクラス52の下位にはMPEGクラス54、オーディオクラス53の下位にはMPEGAudioクラス55が設けられている。

【0059】ストリームクラス51は属性として、ストリームオブジェクトが対応するコンテンツcontents、ストリームオブジェクトに対応するマルチメディアデータのファイル名fileName及びマルチメディアデータを表示する際の負荷情報loadを有する。

【0060】また、ビデオクラス52は、ビデオデータのフレームレートframeRateの属性、コンテンツクラスでの原点に合わせるためのずれ分すなわち先頭からのスキップ分、つまり開始フレームを格納するstartFrameNo、ビデオデータのフレームの大きさwidth、heightの属性、ピクチャレートの異なるオブジェクトを生成し対応するコンテンツのstreamsに設定するnewRate()メソッド、フレームサイズの異なるオブジェクトを生成し対応するコンテンツのstreamsに設定するnewSize()メソッド及びストリームオブジェクトに対応するビデオデータを表示するplay()メソッドを有する。

【0061】また、ビデオクラス52の下位のMPEGクラス54は、MPEG形式のビデオデータを扱うもので、同一コンテンツでフレームレートの異なるビデオストリームを生成するnewRate()メソッド、同一コンテンツでサイズの異なるビデオストリームを生成するnewSize()メソッド及びストリームオブジェクトに対応するビデオデータを表示するplay()メソッドを有する。

【0062】また、オーディオクラス53は、オーディオデータのコーディングレートcodingRateの属性、同一コンテンツでコーディングレートの異なるオーディオストリームを生成するnewRate()メソッド及びストリームオブジェクトに対応するオーディオデータを表示するplay()メソッドを有する。

【0063】また、オーディオクラス53の下位のMPEGAudioクラス55は、MPEG形式のオーディオデータを扱うもので、同一コンテンツでコーディングレートの異なるオーディオストリームを生成するnewRate()メソッド及びストリームオブジェクトに対応するオーディオデータを表示するplay()メソッドを有する。

【0064】次に、ストリームデータのコンテンツへの対応付けの方法について説明する。まず、図7(a)に示すように、コンテンツオブジェクト1に対しストリームオブジェクト1が対応づけられている場合を考える。ここで、ストリームオブジェクト1は、MPEG-1videoフォーマットで、フレームサイズが320×2

40、ピクチャレートが30fps、再生時のCPU11に対する負荷が100%であるとする。

【0065】次に、図7(b)に示すように、ストリームオブジェクト1と異なる品質のストリームデータ、例えば、MPEG-1videoフォーマットで、フレームサイズが320×240、ピクチャレートが15fps、再生時のCPU11に対する負荷が60%のマルチメディアデータと、MPEG-1videoフォーマットで、フレームサイズが160×112、ピクチャレートが30fps、再生時のCPUに対する負荷が35%のマルチメディアデータとを生成し、これらをストリームオブジェクト2、3とする。

【0066】次に、図7(c)に示すように、新たに生成したストリームオブジェクト2、3をコンテンツオブジェクト1に対応づける。これは、ストリームオブジェクト1に対し、それぞれnewRate()メソッド32とnewSize()メソッド33を施すことにより行う。

【0067】以上により、コンテンツオブジェクト1を表示する時に、負荷および品質が異なる3種類のストリームオブジェクト1～3に対応するストリームデータの中から適切なものを選択して用いることができる。

【0068】例えば、動画像と音声とを同期させて出力したり、検索した複数の動画像を同時にウィンドウに表示させたりすることにより、複数のデータを同時に出力する場合、計算機が持つCPU11やI/O22などの資源が有限であるため、資源が足りなくなる可能性がある。

【0069】一方、データを再生する場合に使用する資源とデータの再生品質との間には関連性がある。そして、ある特定の品質を持つデータを再生する場合、その特定の品質を持つデータの再生に要する量の資源が必要である。そのため、資源が足りないまま再生しようとすると、品質の劣化が生じる場合がある。

【0070】特に、動画像の場合、資源が足りない状態で再生すると、フレームの再生が間に合わないため正しい再生レートより遅くなったり、フレームが不定期に飛ばされたりして表示される。また、音声の場合、不定期に音飛びが生じる。

【0071】このため、ビューを再生するときに確保できる資源を用いて再生できる最も品質の良いデータを選択して用いることが望まれる。本実施例では、ビューの再生をする場合、ビューに対応するコンテンツに関連付けられているストリームオブジェクトの属性を見て、適切なデータを選択することができ、限られた資源の範囲内で最も品質の良いデータを再生することができる。

【0072】例えば、データが動画像の場合、データを再生するのに要するCPU11の資源量は、フレームサイズとピクチャレートによって左右される。そのため、デジタル化されているデータに対し、フレームサイズ

17

やピクチャレートを変化させることにより、異なる品質で負荷の異なるデータを新たに生成することができる。

【0073】例えば、ある動画データのパクチャレートを $r\text{ f p s}$ の場合、フレームを n 個置きに選択することにより、パクチャレートを $r/(n+1)\text{ f p s}$ の動画データを生成することができる。そして、パクチャレートを $r/(n+1)\text{ f p s}$ の動画データをコンテンツと対応付けることにより、異なる負荷のデータをコンテンツに追加することが可能である。

【0074】また、既に存在するデータからフレームサイズを小さくしたフレームを生成し、そのフレームを用いてフレームサイズが小さく負荷も小さい（品質も元のデータより劣化した）新たな動画データを生成し、コンテンツと対応付けることができる。

【0075】以上説明したように、本実施例によれば、使用できる資源の範囲内で再生できるデータを選択する自由度を増すことが可能となる。次に、本発明の一実施例によるマルチメディアデータ検索システムにおいて、レートを替えたマルチメディアデータを生成する例を示す。これは、`newRate()` メソッド32を用いて行う。

【0076】図8において、動画データのフレームは一定のインターバルで表示される。この場合、フレームを n 個置きに選択すると、パクチャレートを元のデータより小さいデータを得ることができる。例えば、図8に示すように、パクチャレートを $r\text{ f p s}$ のフレームから1個置きにフレームを選択した場合、パクチャレートを $r/2\text{ f p s}$ のデータを得ることができる。

【0077】図9は、レートを替えたマルチメディアデータの生成装置を示すブロック図である。図9において、MPEGファイル61には、パクチャレートを $r\text{ f p s}$ のMPEG-1 videoデータ36が格納されている。このパクチャレートを $r\text{ f p s}$ のMPEG-1 videoデータをMPEGデコーダ62で復号してフレームデータを作成する。そして、フレームを n 個置きに選択するフレーム選択プログラム63を用いて、このフレームデータを間引く。

【0078】フレーム選択プログラム63により選択されたフレームデータをMPEGエンコーダ64で符号化することにより、パクチャレートを $r/(n+1)\text{ f p s}$ のMPEG-1 videoデータを生成し、MPEGファイル65に格納する。

【0079】そして、このパクチャレートを $r/(n+1)\text{ f p s}$ のMPEG-1 videoデータ36に対応するストリームオブジェクトを生成し、コンテンツオブジェクトに対応づけることにより、コンテンツを異なる品質（図8の例では、パクチャレート）で表現できる。

【0080】次に、本発明の一実施例によるマルチメディアデータ検索システムにおいて、フレームサイズを替えたマルチメディアデータを生成する例を示す。これ

18

は、`newSize()` メソッド33を用いて行い、MPEG-1 videoデータ36から、各画素を $(n-1)$ 個置きに選択することにより、フレームサイズを $1/n$ に縮小する。

【0081】例えば、図10(a)に示すように、元のフレーム71のフレームサイズが c 画素 \times r 画素のMPEG-1 videoデータ36から、図10(b)

(c)に示すように、

$V_{1,0}, V_{3,0}, V_{5,0}, \dots,$
 $V_{0,1}, V_{1,1}, V_{2,1}, \dots,$
 $V_{1,2}, V_{3,2}, V_{5,2}, \dots,$
 $V_{0,3}, V_{1,3}, V_{2,3}, \dots,$
 \vdots

の画素を間引くことにより、縮小したフレーム72のフレームサイズが $c/2$ 画素 \times $r/2$ 画素のMPEG-1 videoデータ36を生成することができる。

【0082】図11は、フレームサイズを替えたマルチメディアデータの生成装置を示すブロック図である。図11において、MPEGファイル81には、フレームサイズが c 画素 \times r 画素のMPEG-1 videoデータ36が格納されている。このフレームサイズが c 画素 \times r 画素のMPEG-1 videoデータ36をMPEGデコーダ82で復号してフレームデータを作成する。このフレームデータの各画素を $(n-1)$ 個置きに選択するサイズ選択プログラム83を用いて間引く。

【0083】サイズ選択プログラム83により縮小されたフレームデータをMPEGエンコーダ84で符号化することにより、フレームサイズが c/n 画素 \times r/n 画素のMPEG-1 videoデータ36を生成し、MPEGファイル85に格納する。

【0084】そして、このフレームサイズが c/n 画素 \times r/n 画素のMPEG-1 videoデータ36に対応するストリームオブジェクトを生成し、コンテンツオブジェクトに対応づけることにより、コンテンツを異なる品質（図10の例では、フレームサイズ）で表現できる。

【0085】なお、上述した実施例では、フレームデータの各画素を $(n-1)$ 個置きに選択してフレームサイズを縮小するようにしていたが、 $n \times n$ 画素の平均値を用いて $1/n$ の大きさに縮小したフレームを作成してもよい。

【0086】次に、本発明の一実施例によるマルチメディアデータ検索システムのビューの機能について説明する。ビューは、コンテンツの一部分をアクセスするために用いるオブジェクトで、コンテンツが動画像や音声など時系列データに関する事象の場合、コンテンツの先頭からの時間とビューの時間長とを用いることにより、ビューのコンテンツに対する位置を表すことができる。また、コンテンツの先頭からの時間とビューの時間長とをビューオブジェクトの属性として記録しておく。なお、

ビュー機能は、コンテンツクラス41で実現される。

【0087】例えば、図12に示すように、コンテンツAの先頭からのスキップする時間skipLengthとビューの時間長timeLengthとを指定することにより、コンテンツAにビューBを設定することができる。

【0088】ビューオブジェクトを実際にユーザに提示する場合、ビューに対応するコンテンツ及びコンテンツの先頭からの時間skipLengthとビューの時間長timeLengthとにより、コンテンツにおけるビューの部分を知ることができる。このため、コンテンツの指定した部分を再生できる再生プログラム19を用いることにより、ビュー機能をユーザに提供することができる。

【0089】図13は、ビューオブジェクトとコンテンツオブジェクトとストリームオブジェクトとの関係を示す図である。図13において、objectAはコンテンツオブジェクトであり、objectBはビューオブジェクトであり、objectS1、S2はストリームオブジェクトである。

【0090】そして、コンテンツオブジェクトAのコンテンツの時間長timeLengthは、例えば、3600秒であり、コンテンツを検索するために用いるキーワードの集合keywordsは“news”であり、コンテンツを具象するストリームオブジェクトの集合streamsはS1とS2であり、コンテンツが情報ビューによって切り出された部分コンテンツである場合の元のコンテンツparentは存在せず、コンテンツに対応する部分コンテンツchildsはBであり、コンテンツの先頭からのスキップする時間skipLengthは0秒であり、事象が生じた時刻absStartTimeは1995年9月1日19時00分00秒であるとする。

【0091】この場合、コンテンツオブジェクトAの再生時間長は3600秒であり、コンテンツを表示するためのストリームオブジェクトとしてobjectS1とobjectS2とを有し、ビューオブジェクトとしてobjectBを有している。

【0092】また、ビューオブジェクトBのビューの時間長timeLengthは、例えば、60秒であり、ビューを検索するために用いるキーワードの集合keywordsは“headline”であり、ビューを具象するストリームオブジェクトの集合streamsはS1とS2であり、ビューによって切り出された部分コンテンツの元のコンテンツparentはAであり、ビューに対応する部分コンテンツchildsは存在せず、コンテンツの先頭からのスキップする時間skipLengthは180秒であり、事象が生じた時刻absStartTimeは1995年9月1日19時03分00秒であるとする。

【0093】この場合、ビューオブジェクトBは、コンテンツオブジェクトAの先頭から180秒の時点からの60秒間に対応している。また、ビューを表示するためのストリームオブジェクトとしてobjectS1とobjectS2とを有している。

【0094】また、ストリームオブジェクトS1の対応するコンテンツcontentsはAであり、ストリームオブジェクトS1に対応する実マルチメディアデータのファイル名fileNameはusr/video/30fps.mpeg、マルチメディアデータを表示する際の負荷loadは35であり、フレームレートframeRateは30であり、コンテンツクラスでの原点に合わせるためのずれ分startFrameNoは存在せず、フレーム幅widthは160であり、フレーム高さheightは112であるとする。

【0095】この場合、ストリームオブジェクトS1は、コンテンツオブジェクトAの実マルチメディアデータに対応し、フレームサイズが160×112画素(PIXEL)、ピクチャレートが30fpsである。

【0096】また、ストリームオブジェクトS2の対応するコンテンツcontentsはAであり、ストリームオブジェクトS2に対応する実マルチメディアデータのファイル名fileNameはusr/video/10fps.mpeg、マルチメディアデータを表示する際の負荷loadは15であり、フレームレートframeRateは10であり、コンテンツクラスでの原点に合わせるためのずれ分startFrameNoは存在せず、フレーム幅widthは160であり、フレーム高さheightは112であるとする。

【0097】この場合、ストリームオブジェクトS2は、コンテンツオブジェクトAの実マルチメディアデータに対応し、フレームサイズが160×112PIXEL、ピクチャレートが10fpsである。

【0098】図14は、ビューオブジェクトの再生処理を示すフローチャートである。まず、ステップS1に示すように、ビューオブジェクトとビューオブジェクトの表示開始時刻とを入力する。

【0099】次に、ステップS2に示すように、ビューオブジェクトの属性parentから、ビューオブジェクトに対応するコンテンツオブジェクトを得る。次に、ステップS3に示すように、コンテンツオブジェクトの属性streamsから、適当なストリームオブジェクトを選択する。

【0100】次に、ステップS4に示すように、選択したストリームオブジェクトに対応するストリームデータを、ビューオブジェクトの属性であるskipLengthの時点から、ビューの時間長timeLengthの間再生する。

【0101】以上のように、ビューを実現するために、与えられた連続メディアデータに対し、指定された

10

20

30

40

50

箇所から再生できなければならない。動画像の場合、先頭からのフレーム数などを用いることにより、音声の場合、先頭からの再生時間などを用いることにより、マルチメディアデータの再生開始場所を指定することができる。

【0102】例えば、MPEG-1 videoフォーマットでは、数枚のフレームを符号化したデータを1まとまりにしたGOP (Group of Pictures) という単位を用いている。

【0103】そして、MPEG-1 videoデータ36を任意のフレームから再生するために、先頭からヘッダ情報を解析しながら所定のフレームまで走査し、表示を開始する方法がある。

【0104】しかし、この方法は、データが膨大になった時も先頭から走査するため効率が悪い。一方、指定されたフレームが、どのGOPに含まれるか、さらにGOP内のフレームの何番目(オフセット)であるかを知ることができれば再生を開始することができる。

【0105】そのため、MPEG-1 videoデータ36の場合、ストリームオブジェクトの属性として、予めデータ内のGOPヘッダの開始バイトと、GOP内で最初に再生されるフレーム番号との対応を調べ、対応表として保持しておく。そして、データのあるフレームから再生する場合、指定されたフレーム番号以下のもので最大のフレーム番号をその対応表から取得し、その最大のフレーム番号と指定フレーム番号との差をオフセットとし、該当GOPの開始バイトを得る。

【0106】また、定符号化レートのMPEG-1 audioデータ37の場合、ストリームオブジェクトの属性として、データの符号化レートを保持する。そして、データの途中からアクセスする場合、スキップする時間長と符号化レートとを用いることにより、データのアクセス開始箇所を計算する。

【0107】例えば、MPEG-1 audioデータ36は、データを符号化したパケットの連なりから構成されており、MPEG-1 audioデータ37には、ほぼ同じ長さのパケットから構成されるモードがある。そして、データの符号化レートがRbps (bit per second) で、再生をスキップする秒数がSの場合、先頭からRS/8byteの位置に一番近いヘッダを検索し、そのパケットから再生を開始する。

【0108】なお、1つのパケットで再生される時間長をTとすると、最大T/2の時間のずれを生じる。次に、本発明の一実施例によるマルチメディアデータ検索システムにおけるスケジューラ8について説明する。

【0109】データベースの検索結果をユーザに提示したり、検索結果に基づいて時系列データの表示を行う場合、複数のビューが同時に再生される場合がある。しかし、時間ごとのビューの多重度を考慮せずそれぞれのビューを独立させたまま、表示に必要な資源と使用するデ

ータとを選択した場合、ある時間に着目した時の負荷が過負荷になってしまう場合がある。そして、過負荷になった場合、不定期にフレーム落ちが発生したり、同期が外れるなどの問題が生じる。

【0110】そのため、1個以上のビューに対し、それぞれの表示開始時刻を指定して表示を指示した場合、ビューと表示開始時刻とを把握しビューに対応するデータから適当なものを選択し、指定された表示開始時刻から表示するようにスケジューラ8を用いて管理する。

【0111】このスケジューラ8は、ビューの再生開始時刻とビューの再生時間長とから、ビューの再生が重なる時間帯を計算する。そして、それぞれの時間帯ごとに、ビュー(コンテンツ)に対応するデータの負荷量を考慮し、負荷量が100%を越えないように使用するデータを選択する。

【0112】例えば、MPEG-1 videoデータ36の場合、I、P、B各フレームの内、処理に一番時間がかかるのは、Iフレームである。そのため、他のビューの再生と重なった場合、正しいピクチャレートで再生するために、Iフレームを処理する時間を確保する必要がある。そこで、再生を行うWSでIフレームを指定レートで表示するのに必要な単位時間あたりの処理時間を負荷量として用いることができる。例えば、Iフレームの処理時間が0.5秒のものは、単位時間が1秒であるのでその負荷を50%とする。

【0113】図15は、Iフレームを用いて負荷量を計算する方法を示すフローチャートである。まず、ステップS11に示すように、MPEG-1 videoデータ36を再生しながら、Iフレームの処理時に、復号及び表示に要する時間長を測定する。

【0114】次に、ステップS12に示すように、復号及び表示に要する時間長の平均を計算し、単位時間で割った値を負荷量とする。スケジューラ18が、コンテンツ16に対するデータを選択するときの順番としては、例えば、対象とするコンテンツ16に対し異なるフレームサイズ及びピクチャレートの動画像データが存在する場合、フレームサイズの大きい方を優先して選択する。また、同じフレームサイズでピクチャレートが異なるものが存在する場合、ピクチャレートの大きい方を優先して選択する。

【0115】また、時間帯で重なるビューオブジェクト間で各時間帯ごとに負荷の調整を行う。例えば、ある時間帯において、負荷の総和が100%を越える場合、上述した順序づけ方法などを用いて決めた順序により候補となるデータを選択し、負荷の総和が100%以下になるようにする。

【0116】その際、スケジューラ18は、あるビューを再生する場合、同一サイズのフレームを採用し、再生されている間にフレームサイズが変化しないようにデータを選択する。すなわち、負荷の調整によるデータの選

択の結果、ある時間帯において、フレームサイズが小さいデータがあるビューに対して採用しなければならない場合、そのビューに対しては、他の時間帯で用いられているフレームサイズも小さいものに変更し、データの選択をやり直すようにする。

【0117】このことにより、あるビューに着目した場合、フレームサイズは各時間帯で一定で、ピクチャレートのみが各時間帯で変化するようにすることができる。そして、同一時間帯の中ではピクチャレートは一定であるので、不定期にフレームが抜けるより見やすくなる。

【0118】また、スケジューラ18に対してコンテンツ16の表示場所が指定されていない場合、スケジューラ18は再生データの時間帯の重なりを考慮して表示場所を決定する。例えば、フレームサイズの異なる動画像を表示する場合、一番大きいフレームに合わせ、ウィンドウシステム35上で領域を区切る。そして、区切られた領域に順番にビューオブジェクトを割り当てる。なお、ある時間帯においてビューの再生が終了した領域が存在する場合、その時間帯以降のビューにその領域を割り当てる。

【0119】以上の動作により、スケジューラ18はビューオブジェクトごとに、ウィンドウシステム35上でウィンドウを作成する位置、フレームサイズ、複数のデータに対しそれぞれ、データのファイル名、データの再生開始時刻、ピクチャレート、データ上の部分アクセスを開始する位置、データの再生時間長を出力する。これらのパラメータは再生プログラム19に読み込まれる。

【0120】次に、本発明の一実施例によるスケジューラ18の動作について説明する。マルチメディアデータベースでは、検索結果を表示するなどの場合、複数のビューオブジェクトを組にしてユーザに提示する。これらのビューオブジェクトを提示するため、それに対応するストリームオブジェクトの中から適切なものをスケジューラ18により選択する。以下の実施例では、動画像ビューオブジェクトを対象とした例を説明する。

【0121】図16は、本発明の一実施例によるスケジューラ18の動作を示すフローチャートである。まず、ステップS21に示すように、スケジューラ18は、複数の<ビューオブジェクト、表示開始時刻>対を入力する。この<ビューオブジェクト、表示開始時刻>対は、例えば、スクリプトをパース（解析）し、各ビューオブジェクトに対する表示開始時刻を計算することにより得られる。

【0122】次に、ステップS22に示すように、各ビューオブジェクトに対し、ビューオブジェクトの再生時間長の属性値をオブジェクトベース13から取得する。次に、ステップS23に示すように、各ビューオブジェクトに対し、ビューオブジェクトに対応する複数のストリームオブジェクトのフレームサイズ（width、height）、ピクチャレート（frameRate

e）、負荷（load）の属性値をオブジェクトベース13から取得する。

【0123】次に、ステップS24に示すように、ビューオブジェクトに対応するストリームオブジェクトに優先順位を付ける。優先順位を付ける方法は、例えば、ストリームオブジェクトをフレームサイズの大きい順に並べ、同じサイズのデータがある場合、ピクチャレートの大きい順に並べる。この優先順位に従って、1から順に数字（識別子）をストリームオブジェクトに付けておく。

【0124】次に、ステップS25に示すように、ビューオブジェクトの時間に対する多重度が変化する区間を検出する。次に、ステップS26に示すように、各ビューオブジェクトの開始時刻と終了時刻（開始時刻と再生時間長とから求められる）とを用いて時間を区切り、区切られた各時間帯で再生されるビューオブジェクトを求め、時間帯ごとに属するビューオブジェクトの一覧表を作る。

【0125】次に、ステップS27に示すように、各時間帯ごとに、それに属するビューオブジェクトとそれに対するストリームオブジェクトの識別子（初期値は1）を記録する。

【0126】次に、ステップS28に示すように、ビューオブジェクトが属する一番過去の時間帯に着目する。次に、ステップS29に示すように、ビューオブジェクトが属する一番過去の時間帯から順に、各時間帯に属するビューオブジェクトに対して選択されているストリームオブジェクトの負荷の総和を計算する。

【0127】例えば、動画像がMPEG-1 videoデータ36の場合、フレームの種類としてIフレーム、Pフレーム、Bフレームがあり、それぞれ復号に要するコンピュータ資源が変化する。他の動画像と同時に表示する場合、資源が最も多く用いられるフレームで負荷量を計算するのが望ましいと考えられる。MPEG-1 videoデータ36では、Iフレームの復号にCPUの時間が最も多く使われるため、Iフレームの復号時間の平均時間T秒を用いて、T%を負荷量として用いることができる。

【0128】次に、ステップS30に示すように、負荷の合計が100%を越えたかどうかを判断し、負荷の合計が100%を越えた場合、ステップS31に進む。次に、ステップS31に示すように、適当なビューオブジェクトに対して、優先順位を下げるのが可能なストリームオブジェクトがあるかどうかを判断し、優先順位を下げるのが可能なストリームオブジェクトがない場合、ステップS32に進み、再生をあきらめる。

【0129】一方、ステップS31で優先順位を下げるのが可能なストリームオブジェクトがあると判断された場合、ステップS33に進み、適当なビューオブジェクトに対して、選択するストリームオブジェクトの優先

10

20

30

40

50

順位を下げる。ビューオブジェクトの選択方法としては、時間帯で最も負荷の大きいビューオブジェクトを選択することもでき、また、選択肢が最も多く残っているビューオブジェクトを選択することもできる。

【0130】次に、ステップS34に示すように、あるビューオブジェクトに対応するストリームオブジェクトの優先順位が下がり、それまでの時間帯に用いていたフレームサイズが変化したビューオブジェクトがあるかどうか判断する。そして、フレームサイズが変化したビューオブジェクトがない場合、ステップS29に戻る。

【0131】一方、ステップS34でフレームサイズが変化したビューオブジェクトがあると判断された場合、ステップS35に進み、フレームサイズが変化したビューオブジェクトの優先順位をXとする。

【0132】次に、ステップS36に示すように、フレームサイズが変化したビューオブジェクトの全ての時間帯に対し、優先順位をXに変更する。次に、ステップS37に示すように、注目している時間帯より以前の各ビューオブジェクトに対する優先度を、現在設定されているフレームサイズで一番優先度の高いものに変更する。

【0133】次に、ステップS38に示すように、注目する時間帯を一番過去の時間帯に設定し、ステップS29に戻ってストリームオブジェクトの決定をやり直す。一方、ステップS30で負荷の合計が100%を越えていないと判断された場合、ステップS39に進み、まだ処理していない時間帯があるかどうかを判断する。そして、まだ処理していない時間帯がある場合、ステップS39に進み、注目する時間帯を次に進め、ステップS29に戻って、以上の計算を全ての時間帯に対して行う。

【0134】一方、ステップS39でまだ処理していない時間帯がないと判断された場合、ステップS41に進み、各ビューオブジェクトの表示位置を計算する。次に、本発明の一実施例によるスケジューラの動作について、4つのビューオブジェクト1~4を表示する場合を例にとって説明する。

【0135】図17(a)に示すように、ビューオブジェクト1~4の選択可能データは、例えば、優先順位1~6のストリームオブジェクトにより示される。優先順位1のストリームオブジェクトは、フレームサイズが320×240PIXELで、ピクチャレートが30fpsであり、CPU11の負荷は100%となる。

【0136】また、優先順位2のストリームオブジェクトは、フレームサイズが320×240PIXELで、ピクチャレートが15fpsであり、CPU11の負荷は60%となる。

【0137】また、優先順位3のストリームオブジェクトは、フレームサイズが320×240PIXELで、ピクチャレートが10fpsであり、CPU11の負荷は45%となる。

【0138】また、優先順位4のストリームオブジェク

トは、フレームサイズが160×112PIXELで、ピクチャレートが30fpsであり、CPU11の負荷は35%となる。

【0139】また、優先順位5のストリームオブジェクトは、フレームサイズが160×112PIXELで、ピクチャレートが15fpsであり、CPU11の負荷は20%となる。

【0140】また、優先順位6のストリームオブジェクトは、フレームサイズが160×112PIXELで、ピクチャレートが10fpsであり、CPU11の負荷は15%となる。

【0141】これらの優先順位は、上述したように、異なるフレームサイズ及びピクチャレートの場合、フレームサイズの大きい方を優先して選択し、同じフレームサイズでピクチャレートが異なる場合、ピクチャレートの大きい方を優先して選択するという方法で決定する。

【0142】また、図17(b)に示すように、ビューオブジェクト1~4を再生する場合、ビューオブジェクト1~4の時間に関する重なりは、例えば、4個の時間帯1~4に分けられるものとする。この場合、時間帯1から順に負荷を計算しながら使用するデータを決定していく。ただし、同一時間帯において、表示される画像はなるべく品質が同じになるように選択する。

【0143】まず、時間帯1において、ビューオブジェクト4のみが表示されるので、優先順位1のストリームオブジェクトを選択し、フレームサイズが320×240PIXELで、ピクチャレートが30fpsで表示され、CPU11の負荷は100%となる。

【0144】次に、時間帯2において、2つのビューオブジェクト3、4が表示される。この2つのビューオブジェクト3、4を優先順位1のストリームオブジェクトで同時に表示した場合、CPU11の負荷の合計は200%となつて、100%を越えてしまうので、ビューオブジェクト3、4に対し、優先順位3のストリームオブジェクトを選択する。このことにより、フレームサイズが320×240PIXELで、ピクチャレートが10fpsで表示され、CPU11の負荷の合計は95%となる。

【0145】次に、時間帯3において、4つのビューオブジェクト1~4が表示される。これら4つを同時に表示するためには、フレームサイズが320×240PIXELの画像を選択することはできず、フレームサイズを縮小する必要がある。そのため、ビューオブジェクト3、4において、時間帯1、2のストリームオブジェクトの選択をやり直し、優先順位4のストリームオブジェクトを選択する。この場合、フレームサイズが160×112PIXELで、ピクチャレートが30fpsの画像が表示され、時間帯1のCPU11の負荷は35%となり、時間帯2のCPU11の負荷の合計は70%となる。

10

20

30

40

50

【0146】また、時間帯3のビューオブジェクト1に対し、優先順位4のストリームオブジェクトを選択し、ビューオブジェクト2~4に対し、優先順位5のストリームオブジェクトを選択する。この場合、CPU11の負荷の合計は95%となる。

【0147】次に、時間帯4において、ビューオブジェクト4のみが表示される。この場合、時間帯3でフレームサイズが160×112PIXELに縮小され、優先順位が4に設定されている。このため、時間帯4でも同サイズでピクチャレートが30fpsの画像を選択する。この場合、CPU11の負荷は35%となる。

【0148】以上により、ビューオブジェクト3の場合、時間帯2において、フレームサイズが160×112PIXELで、ピクチャレートが30fpsの画像データを、時間帯3において、フレームサイズが160×112PIXELで、ピクチャレートが15fpsの画像データを連続して表示することができる。

【0149】また、ビューオブジェクト4の場合、時間帯1、2において、フレームサイズが160×112PIXELで、ピクチャレートが30fpsの画像データを、時間帯3において、フレームサイズが160×112PIXELで、ピクチャレートが15fpsの画像データを、時間帯4において、フレームサイズが160×112PIXELで、ピクチャレートが30fpsの画像データを連続して表示することができる。

【0150】次に、上述した4つのビューオブジェクト1~4を表示するデータ選択方法の処理経過を順を追って説明する。まず、図18(0)に示すように、全ての時間帯1~4において、ビューオブジェクト1~4のストリームオブジェクトは優先順位1に設定される。この場合、時間帯1におけるCPU11の負荷の合計は100%となり、時間帯2におけるCPU11の負荷の合計は200%となり、時間帯3におけるCPU11の負荷の合計は400%となり、時間帯4におけるCPU11の負荷の合計は100%となる。

【0151】次に、図18(1)に示すように、時間帯1に注目し、CPU11の負荷の合計が100%であるので優先順位の変更を行わず、時間帯2の処理に移る。次に、図18(2)に示すように、時間帯2において、CPU11の負荷の合計が100%を越えているので、時間帯2におけるビューオブジェクト4の優先順位を2に下げる。この場合、時間帯2におけるCPU11の負荷の合計は160%となる。なお、優先順位を下げる事が可能なビューオブジェクトが複数個ある場合、優先順位の高いものから選択し、同じ優先順位のビューオブジェクトが複数個ある場合、ビューオブジェクト4、3、2、1の順に選択する。

【0152】次に、図18(3)に示すように、時間帯2において、CPU11の負荷の合計が依然として100%を越えているので、時間帯2におけるビューオブジ

ェクト3の優先順位を2に下げる。この場合、時間帯2におけるCPU11の負荷の合計は120%となる。

【0153】次に、図18(4)に示すように、時間帯2において、CPU11の負荷の合計が依然として100%を越えているので、時間帯2におけるビューオブジェクト4の優先順位を3に下げる。この場合、時間帯2におけるCPU11の負荷の合計は105%となる。

【0154】次に、図18(5)に示すように、時間帯2において、CPU11の負荷の合計が依然として100%を越えているので、時間帯2におけるビューオブジェクト3の優先順位を3に下げる。この場合、時間帯2におけるCPU11の負荷の合計は90%となり、時間帯2におけるCPU11の負荷の合計が100%以下になったので、時間帯3の処理に移る。

【0155】次に、図18(6)に示すように、時間帯3において、CPU11の負荷の合計が100%を越えているので、時間帯3におけるビューオブジェクト4の優先順位を2に下げる。この場合、時間帯3におけるCPU11の負荷の合計は360%となる。

【0156】次に、図18(7)に示すように、時間帯3において、CPU11の負荷の合計が依然として100%を越えているので、時間帯3におけるビューオブジェクト3の優先順位を2に下げる。この場合、時間帯3におけるCPU11の負荷の合計は320%となる。

【0157】次に、図18(8)に示すように、時間帯3において、CPU11の負荷の合計が依然として100%を越えているので、時間帯3におけるビューオブジェクト2の優先順位を2に下げる。この場合、時間帯3におけるCPU11の負荷の合計は280%となる。

【0158】次に、図18(9)に示すように、時間帯3において、CPU11の負荷の合計が依然として100%を越えているので、時間帯3におけるビューオブジェクト1の優先順位を2に下げる。この場合、時間帯3におけるCPU11の負荷の合計は240%となる。

【0159】次に、図18(10)に示すように、時間帯3において、CPU11の負荷の合計が依然として100%を越えているので、時間帯3におけるビューオブジェクト4の優先順位を3に下げる。この場合、時間帯3におけるCPU11の負荷の合計は225%となる。

【0160】次に、図18(11)に示すように、時間帯3において、CPU11の負荷の合計が依然として100%を越えているので、時間帯3におけるビューオブジェクト3の優先順位を3に下げる。この場合、時間帯3におけるCPU11の負荷の合計は210%となる。

【0161】次に、図18(12)に示すように、時間帯3において、CPU11の負荷の合計が依然として100%を越えているので、時間帯3におけるビューオブジェクト2の優先順位を3に下げる。この場合、時間帯3におけるCPU11の負荷の合計は195%となる。

【0162】次に、図18(13)に示すように、時間

29

帯3において、CPU11の負荷の合計が依然として100%を越えているので、時間帯3におけるビューオブジェクト1の優先順位を3に下げる。この場合、時間帯3におけるCPU11の負荷の合計は180%となる。

【0163】次に、図18(14)に示すように、時間帯3において、CPU11の負荷の合計が依然として100%を越えているので、時間帯3におけるビューオブジェクト4の優先順位を4に下げる。この場合、時間帯3におけるCPU11の負荷の合計は170%となる。また、ビューオブジェクト4の優先順位を3から4に下げるにより、フレームサイズが320×240PIXELから160×112PIXELに変化する。

【0164】次に、図18(15)に示すように、全ての時間帯1～4において、フレームサイズが変化したビューオブジェクト4の優先順位を4に変更し、フレームサイズを160×112PIXELに統一する。この場合、時間帯1におけるCPU11の負荷の合計は35%となり、時間帯2におけるCPU11の負荷の合計は80%となり、時間帯3におけるCPU11の負荷の合計は170%となり、時間帯4におけるCPU11の負荷の合計は35%となる。

【0165】次に、図18(16)に示すように、注目していた時間帯より前の時間帯における優先順位を、現在設定されているフレームサイズの範囲内で最も高いものに変更する。この場合、ビューオブジェクト3の時間帯2における優先順位を3から1に変更する。また、時間帯2におけるCPU11の負荷の合計は135%となる。

【0166】次に、図19(17)に示すように、時間帯1に戻り、処理を続ける。時間帯1において、CPU11の負荷の合計が35%であるので優先順位の変更を行わず、時間帯2の処理に移る。

【0167】次に、図18(18)に示すように、時間帯2において、CPU11の負荷の合計が100%を越えているので、時間帯2におけるビューオブジェクト3の優先順位を2に下げる。この場合、時間帯2におけるCPU11の負荷の合計は95%となり、100%以下であるので、時間帯3の処理に移る。

【0168】次に、図18(19)に示すように、時間帯3において、CPU11の負荷の合計が100%を越えているので、時間帯3におけるビューオブジェクト3の優先順位を4に下げる。この場合、時間帯3におけるCPU11の負荷の合計は160%となる。また、ビューオブジェクト3の優先順位を3から4に下げるにより、フレームサイズが320×240PIXELから160×112PIXELに変化する。

【0169】次に、図18(20)に示すように、ビューオブジェクト3の属する時間帯2において、ビューオブジェクト3の優先順位を4に変更し、フレームサイズを160×112PIXELに統一する。この場合、時間

30

帯2におけるCPU11の負荷の合計は70%となる。

【0170】次に、図18(21)に示すように、注目していた時間帯より前の時間帯における優先順位を、現在設定されているフレームサイズの範囲内で最も高いものに変更する。この場合、時間帯1のビューオブジェクト4の優先順位は4であり、現在設定されているフレームサイズの範囲内で最も高いものなので、優先順位の変更を行わない。

【0171】次に、図18(21)に示すように、時間帯1に戻り、処理を続ける。時間帯1において、CPU11の負荷の合計が35%であるので優先順位の変更を行わず、時間帯2の処理に移る。

【0172】次に、図18(22)に示すように、時間帯2において、CPU11の負荷の合計が70%であるので優先順位の変更を行わず、時間帯3の処理に移る。次に、図18(23)に示すように、時間帯3において、CPU11の負荷の合計が100%を越えているので、時間帯3におけるビューオブジェクト2の優先順位を4に下げる。この場合、時間帯3におけるCPU11の負荷の合計は150%となる。

【0173】次に、図18(24)に示すように、時間帯3において、CPU11の負荷の合計が依然として100%を越えているので、時間帯3におけるビューオブジェクト1の優先順位を4に下げる。この場合、時間帯3におけるCPU11の負荷の合計は140%となる。

次に、図18(25)に示すように、時間帯3において、CPU11の負荷の合計が依然として100%を越えているので、時間帯3におけるビューオブジェクト4の優先順位を5に下げる。この場合、時間帯3におけるCPU11の負荷の合計は125%となる。

【0174】次に、図18(26)に示すように、時間帯3において、CPU11の負荷の合計が依然として100%を越えているので、時間帯3におけるビューオブジェクト3の優先順位を5に下げる。この場合、時間帯3におけるCPU11の負荷の合計は110%となる。

【0175】次に、図18(27)に示すように、時間帯3において、CPU11の負荷の合計が依然として100%を越えているので、時間帯3におけるビューオブジェクト2の優先順位を5に下げる。この場合、時間帯3におけるCPU11の負荷の合計は95%となるので、時間帯4の処理に移る。

【0176】次に、図18(28)に示すように、時間帯4において、CPU11の負荷の合計が35%であるので、優先順位の変更を行わず、処理を終了する。次に、本発明の一実施例による複数の動画像を表示する場合のウィンドウの設定方法について説明する。

【0177】フレームサイズの異なる動画像を表示する場合は、一番大きいフレームサイズに合わせてウィンドウを区切る。そして、ビューオブジェクトが前の時間帯に存在するかどうかを時間帯ごとに調べ、ビューオブジ

エクトが前の時間帯に存在する場合、ビューオブジェクトが存在していた場所に引き続きビューオブジェクトを表示する。ビューオブジェクトが前の時間帯に存在しない場合、注目している時間帯で新たに表示されるビューの場所として用いる。

【0178】例えば、図20(a)に示すように、フレームサイズが160×112PIXELのビューオブジェクト1を時間帯1～3に表示し、フレームサイズが320×240PIXELのビューオブジェクト2を時間帯1に表示し、フレームサイズが160×112PIXELのビューオブジェクト3を時間帯3、4に表示する場合、図20(b)に示すように、表示するビューオブジェクト1～3のうちフレームサイズが一番大きいビューオブジェクト2でウィンドウシステムを区切る。そして、ビューオブジェクト2は、時間帯1にのみ表示されるため、ビューオブジェクト2を表示した後、ビューオブジェクト3を表示する場合、ビューオブジェクト3をビューオブジェクト2が表示されていた場所に表示する。

【0179】図21は、複数の動画像を表示する場合のウィンドウの設定方法を示すフローチャートである。まず、ステップS41に示すように、複数のビューを表示する時の各フレームサイズから縦横の最大値をそれぞれ求める。

【0180】次に、ステップS42に示すように、縦横の最大値を用いてウィンドウを区切り、区切られた各区画に対して順番を決める。次に、ステップS43に示すように、注目する時間帯を一番過去に設定する。

【0181】次に、ステップS44に示すように、該当する時間帯で表示する各ビューに対し、ウィンドウの区画を順に割り振る。次に、ステップS45に示すように、注目する時間帯を次の時間帯に進める。

【0182】次に、ステップS46に示すように、表示するビューが直前の時間帯から引き続いているかどうかを判断し、直前の時間帯から引き続いている場合、ステップS47に進み、直前の時間帯で用いている区画を次の時間帯の区画として割り振る。

【0183】一方、ステップS46で直前の時間帯から引き続いていないと判断された場合、ステップS48に進み、空いている区画のうち、最も若い順番の区画を次の時間帯の区画として割り振る。

【0184】次に、ステップS49に示すように、まだ区画が割り振られていない時間帯があるかどうかを判断し、まだ区画が割り振られていない時間帯がある場合、ステップS45に戻って処理を続け、まだ区画が割り振られていない時間帯がない場合、処理を終了する。

【0185】次に、本発明の一実施例による複数の動画像を表示する場合のソフトウェアによる再生方法について説明する。複数の動画像を同時に再生する場合、ハードウェアによる1台の復号装置では1度に1つの動画像

(ビデオストリーム)しか扱うことができないため、任意の数の動画像を再生することができない。一方、ソフトウェアを用いて再生する場合、任意の数の動画像を再生することができる。

【0186】また、スクリプトなどで幾つかのビューとその再生時刻とを指定した場合、幾つかのビューが同時に再生され、同時に再生される動画像の数が時刻によって変化する。このため、同時に再生される動画像の数の変化に応じて、動画像を再生するための負荷を変化させることにより、コンピュータ資源を有効に使用することができる。

【0187】例えば、スケジューラ18は、あるビューに対し、ビュー(動画像)の同時再生数の変化に応じて再生に使用するファイルを選択する。そして、ウィンドウシステム35に作成した1つのウィンドウに選択したファイルを順に再生し同一のウィンドウに表示することにより、ユーザは、データ自体は独立しているにもかかわらず連続したビューが再生されているものとして見るることができる。

【0188】一方、単にウィンドウを生成して動画像を順に再生するのみでは、動画像の切替え部分で処理に時間がかかり連続して再生しているようには見えない。これは、OS27が再生プログラム19を起動する時に生じる初期化に時間がかかったり、MPEG-videoデータ36に部分アクセスをするための処理に時間がかかったりして、再生画像をウィンドウに連続して出力できないためである。

【0189】このため、再生画像の出力に先立ち、予め必要な前処理を行い、指定された再生開始時刻までsleepして待つことにより、再生画像をウィンドウに連続して出力できるようにする。

【0190】すなわち、再生開始時刻より前に動画像再生プログラム19をビューごとに起動する。そして、動画像再生プログラム19のパラメータとして、ウィンドウシステム35上でウィンドウを作成する位置、フレームサイズ、複数のデータに対しそれぞれ、データのファイル名、データの再生開始時刻、ピクチャレート、データ上の部分アクセスを開始する位置、データの再生時間長を入力する。これらのパラメータはスケジューラ18により生成され、それぞれのデータ20の表示時間帯は連続させる。

【0191】例えば、MPEG-videoデータ36の場合、再生プログラム19は、MPEG-videoデータ36が格納されているファイル名と、ウィンドウシステム35での表示用ウィンドウの位置、再生を開始するGOPヘッダの開始バイト、フレームのオフセット、表示開始時刻、表示フレーム数、ピクチャレートを入力する。そして、再生プログラム19は、データ20の再生開始時刻以前にウィンドウシステム35に表示用ウィンドウを生成し、それぞれのデータ20のシーケン

スヘッダ部分を読み込み、初期化を行う。

【0192】その後、指定されたGOPバイトの所にファイルの読み出しポインタを移動し、そこから始まるGOPを解析し、オフセットで指定されたフレーム分を表示せずに復号のみ行う。この初期化のためのGOPの解析を行う場合、`broken_link`をセットし、直前のGOPに依存しているBフレームを再生しないようにする。そのため、GOPに含まれる最初のBフレームの数だけの誤差が生じる。必要に応じて、Bフレームの数分を補正することができる。

【0193】以上により初期化が終了するので、再生開始時刻まで`sleep`して待つ。表示開始時刻になり、OS27により`sleep`状態から`run`状態に遷移すると、ウィンドウシステム35上の指定位置に生成したウィンドウ上に、復号したフレームを指定ピクチャレートで指定フレーム数だけ指定フレームから表示する。次に、本発明の一実施例による複数の動画像を表示する場合の復号や表示のスキップ方法について説明する。

【0194】上述したスケジューラ18はビューの再生のみを把握してスケジュールを作成する。しかし、WSのOS27はマルチタスクで動いており、デバイスからの割り込みなどにより頻繁に処理が切り替わる。そのため、スケジュールを実行しても期待した分のコンピュータ資源を使えるとは限らず、ビューの再生に遅延が生じる可能性がある。連続メディアを指定したレートで正しく再生するためには、必要に応じて復号や表示をスキップしなければならない。

【0195】例えば、MPEG-1 videoデータ36の場合、ピクチャレートが`rfbs`で、フレームSから表示を行う時、フレーム n の`picture_start_code`を復号した時点で、OS27から現在の時刻 T_1 を取得し、表示開始時刻 T_0 からの時間が $(T_1 - T_0) > (n - S) / r$ で（この場合既に表示するタイミングに間に合っていない）、且つフレーム n がBフレームの場合、そのフレーム n を復号しないようにする。

【0196】一方、Iフレームを復号しない場合、PフレームとBフレームとを復号することができず、Pフレームを復号しないとBフレームを復号することができなくなるため、IフレームとBフレームとの復号は常に行う。

【0197】また、フレーム n の復号が終了した時点において、OS27から現在の時刻 T_2 を取得し、表示開始時刻からの時間が $(T_2 - T_0) > (n - S) / r$ の場合、表示を行わずに即座に次の処理を行う。

【0198】次に、本発明の一実施例によるMPEG-1 videoデータ36の指定位置からの表示方法について説明する。MPEG-1 videoデータ36は、図22(a)に示すように、先頭に`sequence_header`があり、その後、複数のGOP#1、G

OP#2、・・・が続く。`sequence_header`は、先頭に`sequence_header_code`（\$00000100）があり、フレームサイズなどの情報が含まれる。GOPは、先頭に`group_start_code`（\$000001B8）があり、フレームデータが含まれる。

【0199】MPEG-1 videoデータ36の属性として、図22(b)に示すように、MPEGデコーダを用い、`group_start_code`が出現したときの先頭からのバイト数とそれまでに復号したフレーム数（該当GOPに含まれる最小フレームナンバ）とを記録する。

【0200】そして、該当するMPEG-1 videoデータ36に対し、再生開始フレームナンバ n が与えられた場合、属性として記録されている<フレームナンバ、バイト数>対を用い、フレームナンバ n より小さい最も大きいフレームナンバ f を検索する。フレームナンバ n は、フレームナンバ f に対応するバイト数から始まるGOPに含まれる。フレームナンバ n から再生するために $(n - f)$ フレームを表示せずに復号し、次のフレームから表示を開始する。

【0201】図23は、本発明の一実施例によるMPEG-1 videoデータ36の指定フレーム番号からの表示方法を示すフローチャートである。まず、ステップS51に示すように、表示したいMPEG-1 videoデータ36のファイル名F、GOP開始バイトとフレーム番号との対応表T、表示を開始するフレーム番号Aを入力する。

【0202】次に、ステップS52に示すように、対応表Tのフレーム番号欄からフレーム番号Aを越えない最大のフレーム番号Bを検索する。次に、ステップS53に示すように、フレーム番号Bに対応するGOP開始バイトCを対応表Tから得る。

【0203】次に、ステップS54に示すように、MPEG-1 videoデータ36のファイル名Fの先頭部分にあるシーケンスヘッダを読み込み復号する。次に、ステップS55に示すように、ファイル名FのファイルポインタをGOP開始バイトCに移す。

【0204】次に、ステップS56に示すように、GOP開始バイトCの位置から $(A - B)$ 枚分のフレームデータに対し、表示せずに復号のみ行う。次に、ステップS57に示すように、引き続き復号し、得られたフレームデータを表示する。

【0205】次に、本発明の一実施例によるMPEG-1 audioデータ37の指定位置からの再生方法について説明する。MPEG-1 audioデータ37は、図24に示すように、'11111111111111'のヘッダで始まるほぼ同じ長さのパケット#1、パケット#2、パケット#3、・・・から構成されるモードがある。そして、データの符号化レートが`Rbps`で、再生

10

20

30

40

50

35

をスキップする秒数がSの場合、先頭からRS/8byteの位置に1番近いヘッダを検索し、そのパケットから再生を開始する。

【0206】MPEG-1audioデータ37はバイト単位で符号化されているので、先頭からRS/8byteの位置からパケット長の半分の長さL前から2Lバイトの領域に対し、ヘッダパターンにマッチングするかどうかを調べることで検索を行う。

【0207】図25は、本発明の一実施例によるMPEG-1audioデータ37の指定スキップ時間からの再生方法を示すフローチャートである。まず、ステップS61に示すように、表示したいMPEG-1audioデータ37のファイル名F、符号化レートRbps、スキップする時間長S秒を入力する。

【0208】次に、ステップS62に示すように、MPEG-1audioデータ37のファイル名Fのファイルポインタを先頭からRS/8byteの位置に移す。次に、ステップS63に示すように、ファイル名Fのファイルポインタの前後で1番近いフレームヘッダを探し、そのフレームヘッダの先頭にファイルポインタを移す。

【0209】次に、ステップS64に示すように、ファイルポインタの位置からデータを読み込み復号して出力する。次に、本発明の一実施例によるMPEG-1videoデータ36の指定箇所を指定時間再生する方法について説明する。

【0210】まず、MPEG-1videoデータ36が格納されているファイル名、ウィンドウシステム35での表示用ウインドウの位置、再生を開始するGOPヘッダの開始バイト、フレームのオフセット、表示開始時刻、表示フレーム数、ピクチャレートを指定する。

【0211】そして、再生プログラム19が起動されると、最初にウィンドウシステム35に表示するためのウィンドウを1つ生成する。ウィンドウは透明な状態で生成し、ウィンドウ上には表示されない。

【0212】その後、指定されたファイルごとに、シーケンスヘッダ部分を読み込み、初期化を行う。さらに、指定したGOPバイトで指示された場所にファイルのポインタを移動し、オフセットが0よりも大きい場合、オフセットで指定されているフレームを復号する。また、最初に再生するGOPを解析する時に、broken_linkをセットする。

【0213】ファイルごとの初期化が終了した後、現在の時刻をOS27から取得し、表示開始時刻までの差を計算する。そして、その時間長だけnanosleep()システムコールを用いてsleep状態に入る。

【0214】sleep状態が終わった後、表示開始時刻が若い順に初期化した状態から引続き復号を行う。次に、本発明の一実施例による複数の動画の連続表示方法について説明する。

36

【0215】図26(a)に示すように、例えば、ビューオブジェクトに対応するデータ1~3がそれぞれ時間帯1~3に分かれて表示される場合、まず、表示プログラムを立ち上げる。そして、図26(b)に示すように、OS27のプログラム読み込みなどで生じるsetup時間の経過後、ウィンドウを生成する。さらに、表示するデータ1~3のシーケンスヘッダ部分を読み込み、それぞれのデータ1~3の初期化を行う(図26(b)のAの部分)。

10 【0216】その後、最初に表示するデータ1の表示開始時刻まで、sleep状態に入る(図26(b)のBの部分)。そして、sleep状態が終了した後、データ1~3を連続表示する。

【0217】以上説明したように、データ1~3の表示の切り替えの際に生じる初期化処理1~3を、データ1~3の表示開始前に行うことにより、データ1~3の表示開始時刻に余分な処理を行うことを省略でき、データ1~3を滑らかにつないで表示することができる。

20 【0218】図27は、本発明の一実施例による同一ウィンドウに動画像を連続して表示する方法を示すフローチャートである。まず、ステップS71に示すように、1つ以上の動画データ36のファイル名、再生開始時刻、データの部分アクセスする場所、ピクチャレート、再生時間長>を入力する。

【0219】次に、ステップS72に示すように、ウィンドウシステム35上にウィンドウを1つ生成する。次に、ステップS73に示すように、表示するデータを再生開始時刻の順に並べる。

30 【0220】次に、ステップS74に示すように、各データに対し、データの部分アクセスする場所から再生するための初期化を行う。次に、ステップS75に示すように、最初に再生するデータの再生開始時刻まで待つ。

【0221】次に、ステップS76に示すように、再生開始時刻がきたデータをウィンドウに再生する。次に、ステップS77に示すように、まだ再生するデータがあるかどうか判断し、再生するデータがない場合、処理を終了し、再生するデータがある場合、ステップS78に進み、処理を次のデータに移し、ステップS76に戻る。

40 【0222】図28は、本発明の一実施例による同一ウィンドウにMPEG-1videoデータ36を連続して表示する方法を示すフローチャートである。まず、ステップS81に示すように、1つ以上のMPEG-1videoデータ36のファイル名、再生開始時刻、部分アクセスを開始するGOP、ピクチャレート、再生時間長>を入力する。

50 【0223】次に、ステップS82に示すように、ウィンドウシステム35上にウィンドウを1つ生成する。次に、ステップS83に示すように、表示するデータを再生開始時刻の順に並べる。

【0224】次に、ステップS84に示すように、各データに対し、シーケンスヘッダを解析し、指定したGOPにファイルポインタを移動し、オフセット分のフレームを表示せずに復号する。

【0225】次に、ステップS85に示すように、最初に再生するデータの再生開始時刻まで待つ。次に、ステップS86に示すように、再生開始時刻がきたデータをウィンドウに再生する。

【0226】次に、ステップS87に示すように、まだ再生するデータがあるかどうか判断し、再生するデータがない場合、処理を終了し、再生するデータがある場合、ステップS88に進み、処理を次のデータに移し、ステップS86に戻る。

【0227】図29は、本発明の一実施例によるビューを複数のストリームデータを用いて表示する方法を示すフローチャートである。まず、ステップS91に示すように、ビューに対し、スケジューラ18を用いることにより、<動画データファイル名、再生開始時刻、データの部分アクセスする場所、ピクチャレート、再生時間長>を入力する。

【0228】次に、ステップS92に示すように、ウィンドウシステム35上にウィンドウを1つ生成する。次に、ステップS93に示すように、表示するデータを再生開始時刻の順に並べる。

【0229】次に、ステップS94に示すように、各データに対し、データの部分アクセスする場所から再生するための初期化を行う。次に、ステップS95に示すように、最初に再生するデータの再生開始時刻まで待つ。

【0230】次に、ステップS96に示すように、再生開始時刻がきたデータをウィンドウに再生する。次に、ステップS97に示すように、まだ再生するデータがあるかどうか判断し、再生するデータがない場合、処理を終了し、再生するデータがある場合、ステップS98に進み、処理を次のデータに移し、ステップS96に戻る。

【0231】次に、本発明の一実施例によるMPEG-1 videoデータ36を指定レートで表示する方法について説明する。図30において、例えば、ピクチャレートが r fpsのデータを入力し、 $\#S$ フレームから表示を行う場合、 $\#(S+2)$ フレームの`picture_start_code`を復号した時点でOS27のシステムコールを用い、現在の時刻 T_1 を取得する。そして、与えられた表示開始時刻 T_0 からの時間が $(T_1 - T_0) > (\#(S+2) - \#S) / r$ で、且つ $\#(S+2)$ フレームがBフレームの場合、 $\#(S+2)$ フレームの`picture_start_code`の復号の終了した時刻が $\#(S+2)$ フレームを表示すべき時刻から遅れているため、その $\#(S+2)$ フレームの復号をせず、次の処理に移る。

【0232】また、 $\#(S+2)$ フレームの復号が終了

した時点において、OS27から現在の時刻 T_2 を取得し、表示開始時刻 T_0 からの時間が $(T_2 - T_0) >$

$(\#(S+2) - \#S) / r$ の場合、 $\#(S+2)$ フレームの復号が終了した時刻が $\#(S+2)$ フレームを表示すべき時刻から遅れているため、表示を行わない。

【0233】図31は、本発明の一実施例によるMPEG-1 videoデータ36を指定レートで表示する方法を示すフローチャートである。まず、ステップS101に示すように、<MPEG-1 videoデータ36のファイル名、再生開始時刻 T_0 、表示開始フレーム S 、ピクチャレート r >を入力する。

【0234】次に、ステップS102に示すように、 $\#S$ フレームからデータを表示するための初期化を行う。次に、ステップS103に示すように、表示開始フレーム S のフレーム番号 S を変数 n に入れる。

【0235】次に、ステップS104に示すように、再生開始時刻 T_0 まで待つ。次に、ステップS105に示すように、 $\#n$ フレームを表示する。次に、ステップS106に示すように、まだ処理を行っていないフレームがあるかどうかを判断し、処理を行っていないフレームがない場合、処理を終了する。

【0236】一方、ステップS106で処理を行っていないフレームがあると判断された場合、ステップS107に進み、変数 n を1つ増加させる。次に、ステップS108に示すように、 $\#n$ フレームの`picture_start_code`を復号する。

【0237】次に、ステップS109に示すように、OS27から現在の時刻 T_1 を得る。次に、ステップS110に示すように、 $(T_1 - T_0) > (n - S) / r$ で、且つ $\#n$ フレームがBフレームであるかどうかを判断し、 $(T_1 - T_0) > (n - S) / r$ で、且つ $\#n$ フレームがBフレームである場合、ステップS106に戻る。

【0238】一方、ステップS110において、 $(T_1 - T_0) > (n - S) / r$ で、且つ $\#n$ フレームがBフレームでないと判断された場合、ステップS111に進み、 $\#n$ フレームの復号を行う。

【0239】次に、ステップS112に示すように、OS27から現在の時刻 T_2 を得る。次に、ステップS113に示すように、 $(T_2 - T_0) > (n - S) / r$ であるかどうかを判断し、 $(T_2 - T_0) > (n - S) / r$ である場合、ステップS106に戻る。

【0240】一方、ステップS113において、 $(T_2 - T_0) > (n - S) / r$ でないと判断された場合、ステップS114に進み、 $(n - S) / r - T_2$ の間待つ。以上説明したように、本発明の実施例によれば、マルチメディアデータベースにおいてマルチメディアデータを直接扱うのではなく、ビュー（コンテンツ）という論理的な単位で扱うことができ、ビューを表示する際に品質及び負荷がコンピュータ資源に適應するようにマル

チメディアデータを選択することができるため、マルチメディアデータを柔軟に扱うことが可能となる。

【0241】

【発明の効果】以上説明したように、請求項1の発明によれば、マルチメディアデータを事象ごとに管理することにより、1つの事象に複数のマルチメディアデータに対応させることができ、1つの事象に対応する異なる品質のマルチメディアデータを用意しておくことにより、マルチメディアデータを再生するために必要な負荷を制御することができる。

【0242】また、請求項2の発明によれば、1つの事象に対応する複数のマルチメディアデータに共通の識別子に対応させることにより、1つの事象に対応する複数のマルチメディアデータを容易に取り出すことができる。

【0243】また、請求項3の発明によれば、マルチメディアデータを表示する際の処理に必要な負荷量に基づいて、共通の識別子を有する複数のマルチメディアデータの中から1つを選択することにより、マルチメディアデータを再生するために必要な負荷を制御しながら、1つの事象に対応する複数のマルチメディアデータから適切なものを容易に取り出すことができる。

【0244】また、請求項4の発明によれば、コンテンツに対応づけられている1つ以上の品質の異なるデータの中から適切なものを選択することにより、同一のコンテンツで異なる品質のマルチメディアデータから適切なものを選択することができ、マルチメディアデータを再生するために必要な負荷を制御することができる。

【0245】また、請求項5の発明によれば、ピクチャレートが r フレーム/秒の動画データに対し、フレームを n 個置きに選択することにより、同一のコンテンツで異なる品質のマルチメディアデータを容易に生成することができる。

【0246】また、請求項6の発明によれば、フレームサイズの小さい画像データを生成することにより、同一のコンテンツで異なる品質のマルチメディアデータを容易に生成することができる。

【0247】また、請求項7の発明によれば、マルチメディアデータの内容の一部を抽出してビューを設定することにより、マルチメディアデータの一部分を容易に検索することができる。

【0248】また、請求項8の発明によれば、ビューに対応するマルチメディアデータの再生処理に必要な負荷量に基づいて、再生に使用するマルチメディアデータを選択することにより、同一のコンテンツで異なる品質のマルチメディアデータの一部分から適切なものを選択することができ、マルチメディアデータの一部分を再生するために必要な負荷を制御することができる。

【0249】また、請求項9の発明によれば、MPEGデータのIフレームを指定レートで表示するのに必要な

単位時間当たりの処理時間を負荷量として用いることにより、マルチメディアデータを再生するために必要な負荷量を短時間で容易に求めることができる。

【0250】また、請求項10の発明によれば、時間帯ごとの負荷量が100%を越えないように、再生に使用するマルチメディアデータを選択することにより、マルチメディアデータの再生に用いる資源の範囲内でマルチメディアデータの再生を行うことができる。

10 【0251】また、請求項11の発明によれば、フレームサイズが大きい方を優先して選択し、同一フレームサイズ内では、ピクチャレートの大きい方を優先して選択することにより、マルチメディアデータの再生に用いる資源の範囲内で最適な品質のマルチメディアデータの再生を行うことができる。

【0252】また、請求項12の発明によれば、同一のビューに対して同一サイズのフレームを採用することにより、再生の途中にフレームサイズが変化して、画面が見にくくなることを回避できる。

20 【0253】また、請求項13の発明によれば、ビューのフレームサイズを一致させることにより、再生の途中にフレームサイズが変化して、画面が見にくくなることを防止できる。

【0254】また、請求項14の発明によれば、ビューに対応する各ストリームデータに優先順位を付与し、負荷量の合計が所定の値を越える場合、負荷量の低い優先順位のものに変更することにより、複数のビューを同時に表示する場合において、再生の途中にフレームサイズが変化して、画面が見にくくなることを防止しながら、マルチメディアデータの再生に用いる負荷量を資源の範囲内に抑えることができる。

30 【0255】また、請求項15の発明によれば、時間帯ごとに再生時のピクチャレートを変化させることにより、マルチメディアデータの一部分を再生するために必要な負荷を制御することができる。

【0256】また、請求項16の発明によれば、マルチメディアデータを表示するウィンドウ同士が重ならないように表示場所を決定することにより、複数のビューが重なったまま同時に再生されることを防止できる。

40 【0257】また、請求項17の発明によれば、オフセット分のフレームを表示することなく、MPEG-1 videoデータを復号し、指定フレームからMPEG-1 videoデータの再生を開始することにより、MPEG-1 videoデータを指定フレームから正確に再生することができる。

【0258】また、請求項18の発明によれば、MPEG-1 audioデータのアクセス開始箇所に最も近いフレームヘッダを検索し、MPEG-1 audioデータの再生を開始することにより、MPEG-1 audioデータを途中から正確に再生することができる。

50 【0259】また、請求項19の発明によれば、各マル

チメディアデータの再生される時間帯が重ならず、且つ連続している場合、各マルチメディアデータを再生開始時刻よりウィンドウ上に表示を開始することにより、複数のマルチメディアデータを指定した再生開始時刻から連続して再生することができる。

【0260】また、請求項20の発明によれば、各MPEG-1 videoデータの再生開始時刻以前に初期化を行うことにより、各MPEG-1 videoデータを指定した再生開始時刻になった時点で即座に再生することができ、複数のMPEG-1 videoデータを連続して再生することができる。

【0261】また、請求項21の発明によれば、表示開始時刻までに復号や表示が間に合わないMPEG-1 videoデータの復号や表示をスキップすることにより、資源を有効に使うことができる。

【0262】また、請求項22の発明によれば、同一のコンテンツの複数のマルチメディアデータから適切なものを選択することにより、マルチメディアデータを再生するために必要な負荷を制御することができる。

【0263】また、請求項23の発明によれば、符号化された複数のマルチメディアデータを予め復号しておくことにより、復号に要する時間に影響されることなく、複数のマルチメディアデータを連続して表示することができる。

【図面の簡単な説明】

【図1】本発明の一実施例によるマルチメディアデータ検索システムの機能的な構成を示すブロック図である。

【図2】本発明の一実施例によるマルチメディアデータ検索システムの概略構成を示すブロック図である。

【図3】本発明の一実施例によるマルチメディアデータ検索システムの概略構成を示すブロック図である。

【図4】本発明の一実施例によるコンテンツオブジェクトとストリームオブジェクトとマルチメディアデータとの関係を示す図である。

【図5】本発明の一実施例によるコンテンツオブジェクトのクラス構造を示す図である。

【図6】本発明の一実施例によるストリームオブジェクトのクラス構造を示す図である。

【図7】本発明の一実施例によるコンテンツオブジェクトとストリームオブジェクトとの対応づけの方法を示す図である。

【図8】本発明の一実施例によるマルチメディアデータ検索システムのフレームの選択方法を示す図である。

【図9】本発明の一実施例によるピクチャレートを下げたマルチメディアデータ作成装置の構成を示す図である。

【図10】本発明の一実施例によるマルチメディアデータ検索システムのフレームサイズの縮小方法を示す図である。

【図11】本発明の一実施例によるフレームサイズを小

さくしたマルチメディアデータ作成装置の構成を示す図である。

【図12】本発明の一実施例によるコンテンツとビューとの時間関係を示す図である。

【図13】本発明の一実施例によるコンテンツとビューとの関係を示す図である。

【図14】本発明の一実施例によるビューオブジェクトの再生方法を示すフローチャートである。

【図15】本発明の一実施例によるIフレームを用いる負荷量の計算方法を示すフローチャートである。

【図16】本発明の一実施例によるスケジューラの動作方法を示すフローチャートである。

【図17】本発明の一実施例によるスケジューラの選択可能データとスケジュール結果とを示すフローチャートである。

【図18】本発明の一実施例によるスケジューラの選択処理の経過を示す図である。

【図19】本発明の一実施例によるスケジューラの選択処理の経過を示す図である。

【図20】本発明の一実施例によるスケジューラの表示データの画面配置方法を示す図である。

【図21】本発明の一実施例によるウィンドウ表示の位置の決定方法を示すフローチャートである。

【図22】本発明の一実施例によるマルチメディアデータ検索システムのGOPテーブルの作成方法を示す図である。

【図23】本発明の一実施例によるMPEG-1ビデオデータの指定位置からの表示方法を示すフローチャートである。

【図24】本発明の一実施例によるMPEG-1オーディオフォーマットの構成を示す図である。

【図25】本発明の一実施例によるMPEG-1オーディオデータの指定位置からの表示方法を示すフローチャートである。

【図26】本発明の一実施例によるマルチメディアデータの表示処理の時間遷移を示す図である。

【図27】本発明の一実施例によるMPEG-1ビデオデータの同一ウィンドウに対する連続表示方法を示すフローチャートである。

【図28】本発明の一実施例によるMPEG-1ビデオデータの同一ウィンドウに対する連続表示方法を示すフローチャートである。

【図29】本発明の一実施例によるビューの複数データを用いる連続表示方法を示すフローチャートである。

【図30】本発明の一実施例によるフレーム表示タイミングを示す図である。

【図31】本発明の一実施例によるMPEG-1ビデオデータのフレーム表示方法を示すフローチャートである。

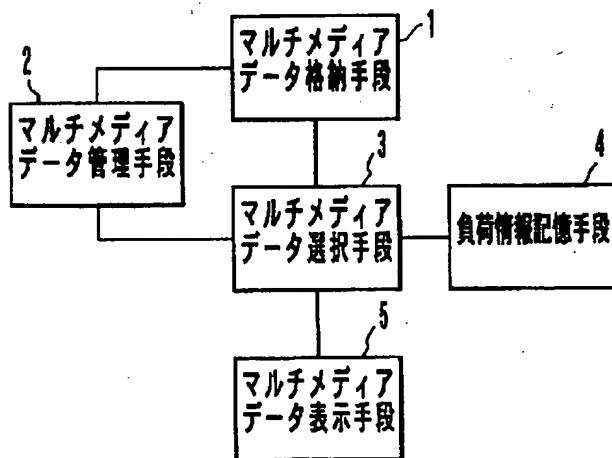
【符号の説明】

43

- 1 マルチメディアデータ格納手段
- 2 マルチメディアデータ管理手段
- 3 マルチメディアデータ選択手段
- 4 負荷情報記憶手段
- 5 マルチメディアデータ表示手段
- 11 CPU
- 12 記憶装置
- 13 オブジェクトベース
- 14 クラス
- 15 メソッド
- 16 コンテンツ
- 17 ストリーム
- 18 スケジューラ
- 19 再生プログラム
- 20 メディアデータ
- 21 メモリ
- 22 入出力装置
- 23 キーボード
- 24 CRT
- 25 スピーカ
- 26 データベースエンジン
- 27 オペレーティングシステム
- 28 keyword () メソッド
- 29 setView () メソッド
- 30 newVideo () メソッド

【図1】

本発明の一実施例によるマルチメディアデータ
検索システムの機能的な構成を示すブロック図

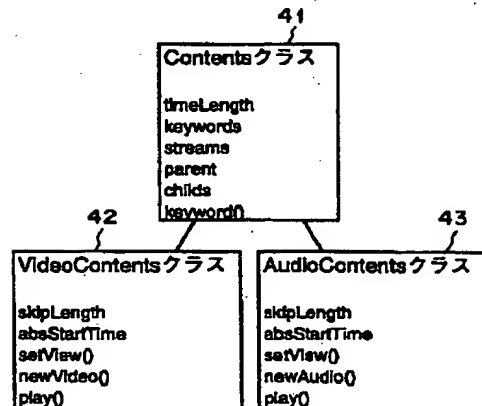


44

- 31 newAudio () メソッド
- 32 newRate () メソッド
- 33 newSize () メソッド
- 34 play () メソッド
- 35 ウィンドウシステム
- 36 MPEG1ビデオデータ
- 37 MPEG1オーディオデータ
- 38 コンテンツオブジェクト
- 39 a ~ 39 c ストリームオブジェクト
- 10 40 a ~ 40 c マルチメディアデータ
- 41 コンテンツクラス
- 42 ビデオコンテンツクラス
- 43 オーディオコンテンツクラス
- 23 イメージコンテンツクラス
- 24 テキストコンテンツクラス
- 51 ストリームクラス
- 52 ビデオクラス
- 53 オーディオクラス
- 54 MPEGクラス
- 20 55 MPEGオーディオクラス
- 61、65、81、85 MPEGファイル
- 62、82 MPEGデコーダ
- 63、83 フレーム選択プログラム
- 64、84 MPEGエンコーダ
- 71、72 フレーム

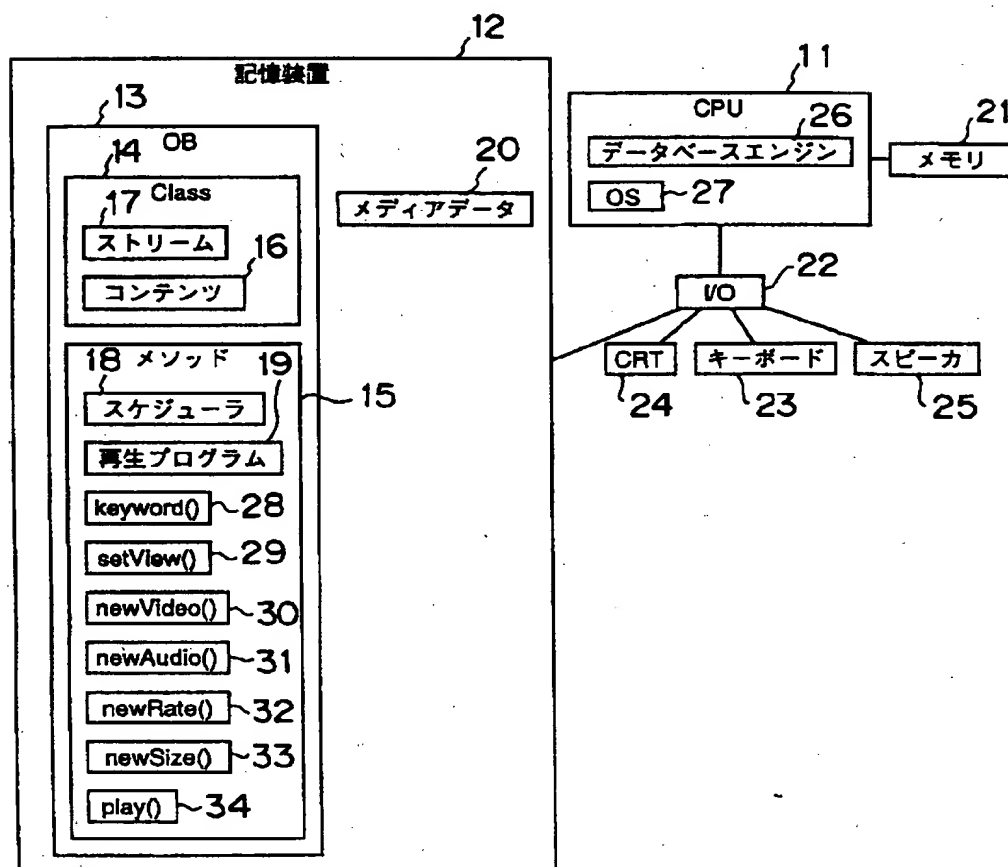
【図5】

本発明の一実施例による
コンテンツオブジェクトのクラス構造を示す図



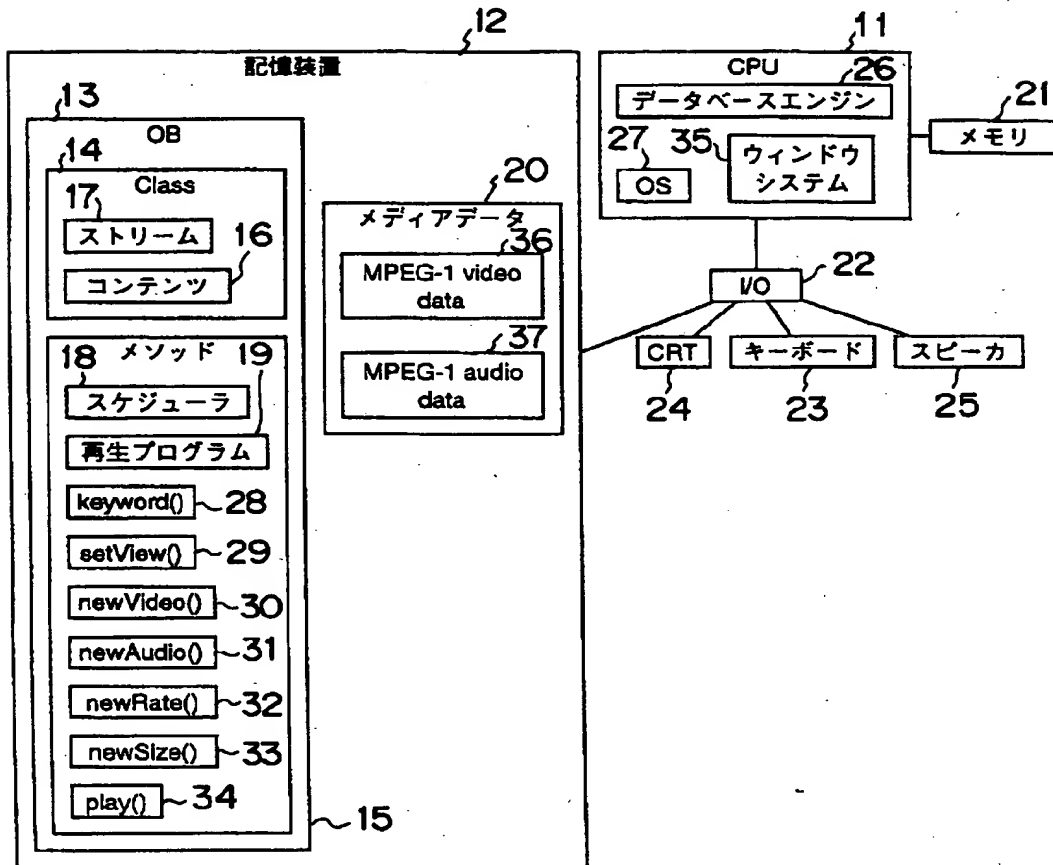
【図 2】

本発明の一実施例によるマルチメディアデータ
検索システムの概略構成を示すブロック図



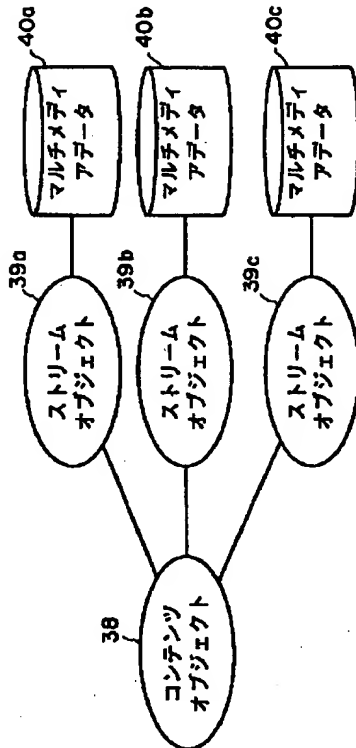
【図3】

本発明の一実施例によるマルチメディアデータ
検索システムの概略構成を示すブロック図



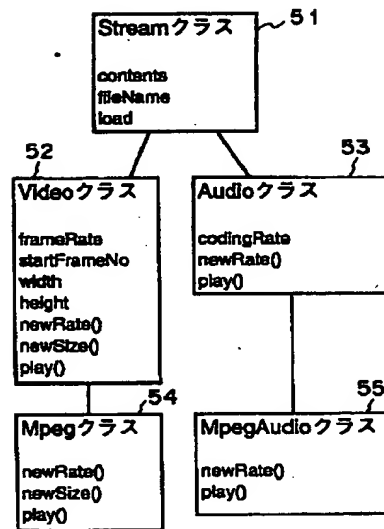
【図4】

本発明の一実施例によるコンテンツオブジェクトと
ストリームオブジェクトとマルチメディアデータの
関係を示す図



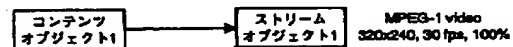
【図6】

本発明の一実施例による
ストリームオブジェクトのクラス構造を示す図

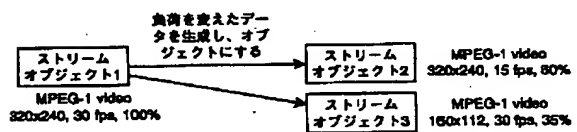


【図7】

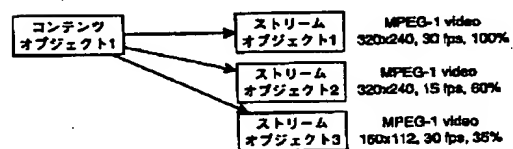
本発明の一実施例によるコンテンツオブジェクトと
ストリームオブジェクトとの対応づけの方法を示す図



(a) 初期状態



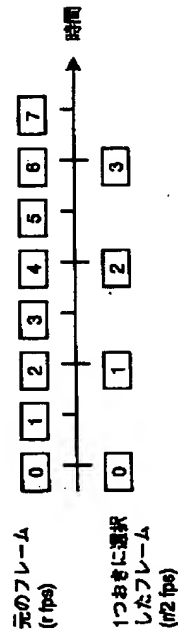
(b) データ生成



(c) コンテンツへ対応づけ

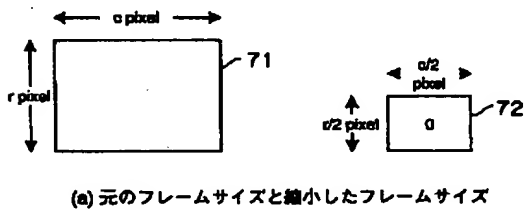
【図8】

本発明の一実施例によるマルチメディアデータ検索システムのフレームの選択方法を示す図



【図10】

本発明の一実施例によるマルチメディアデータ検索システムのフレームサイズの縮小方法を示す図



(b) 元のフレーム(一部)

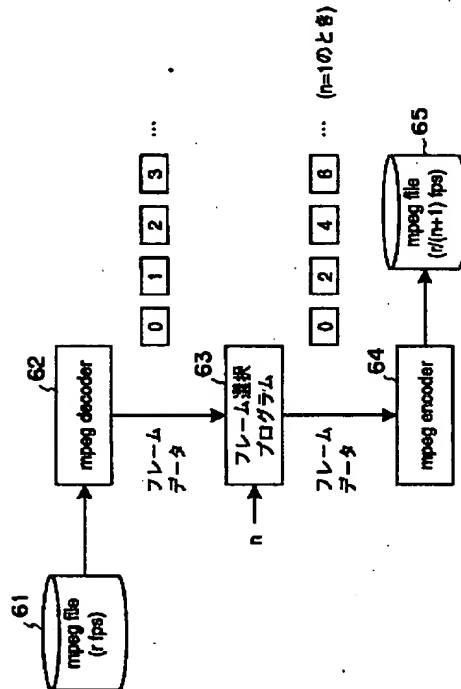
$V_{0,0}$	$V_{1,0}$	$V_{2,0}$	$V_{3,0}$
$V_{0,1}$	$V_{1,1}$	$V_{2,1}$	$V_{3,1}$
$V_{0,2}$	$V_{1,2}$	$V_{2,2}$	$V_{3,2}$

(c) 縮小したフレーム(一部)

$V_{0,0}$	$V_{2,0}$	$V_{4,0}$
$V_{0,2}$	$V_{2,2}$	$V_{4,2}$
$V_{0,4}$	$V_{2,4}$	$V_{4,4}$

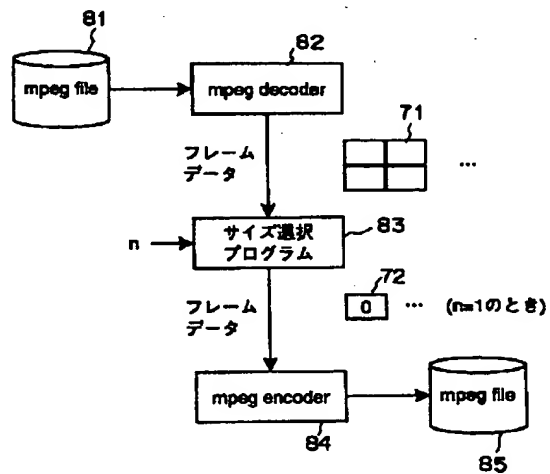
【図9】

本発明の一実施例によるピクチャレートを下げたマルチメディアデータ作成装置の構成を示す図



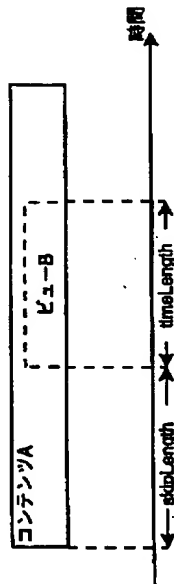
【図11】

本発明の一実施例によるフレームサイズを小さくしたマルチメディアデータ作成装置の構成を示す図



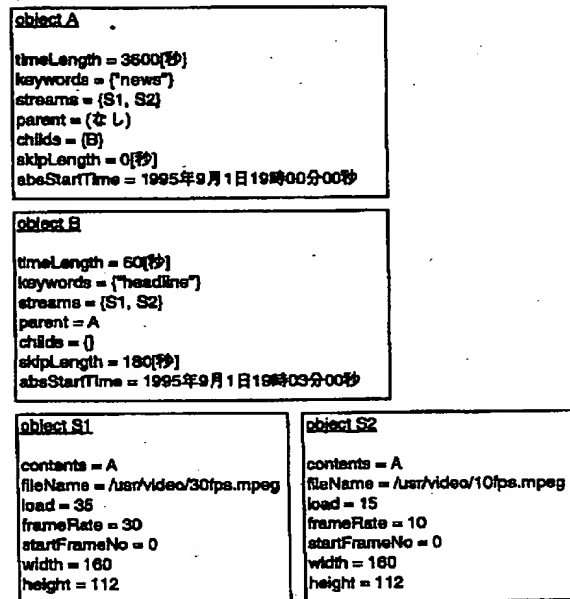
【図 1 2】

本発明の一実施例による
コンテンツとビューとの時間関係を示す図



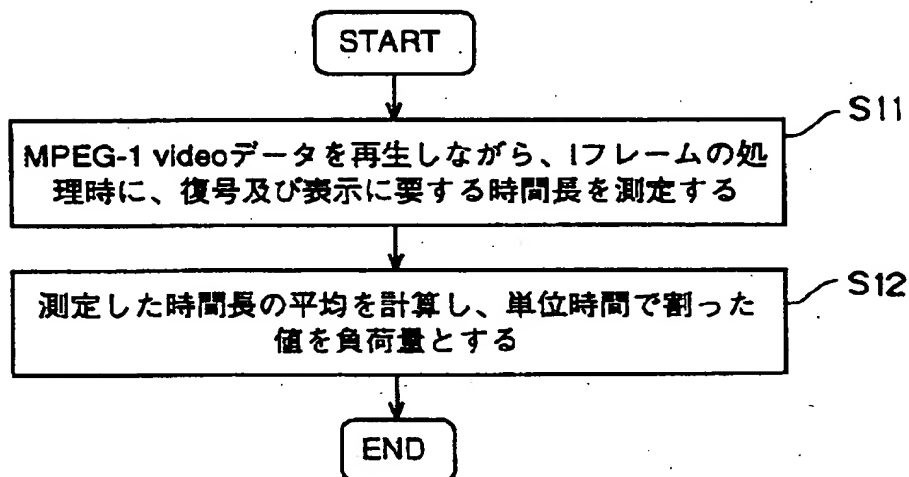
【図 1 3】

本発明の一実施例による
コンテンツとビューとの関係を示す図



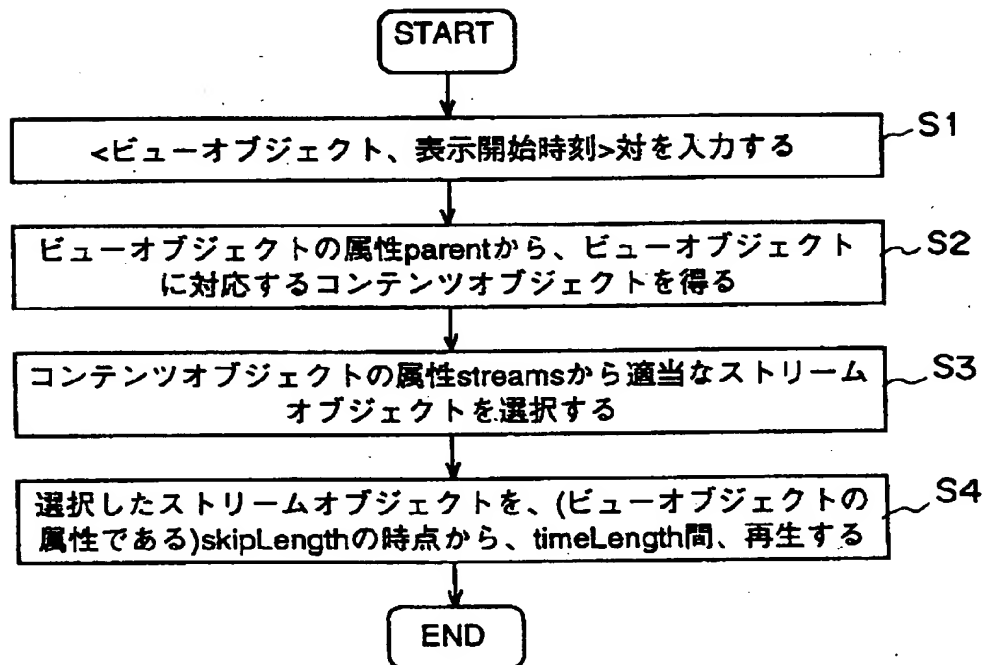
【図 1 5】

本発明の一実施例による I フレームを用いる
頁荷量の計算方法を示すフローチャート



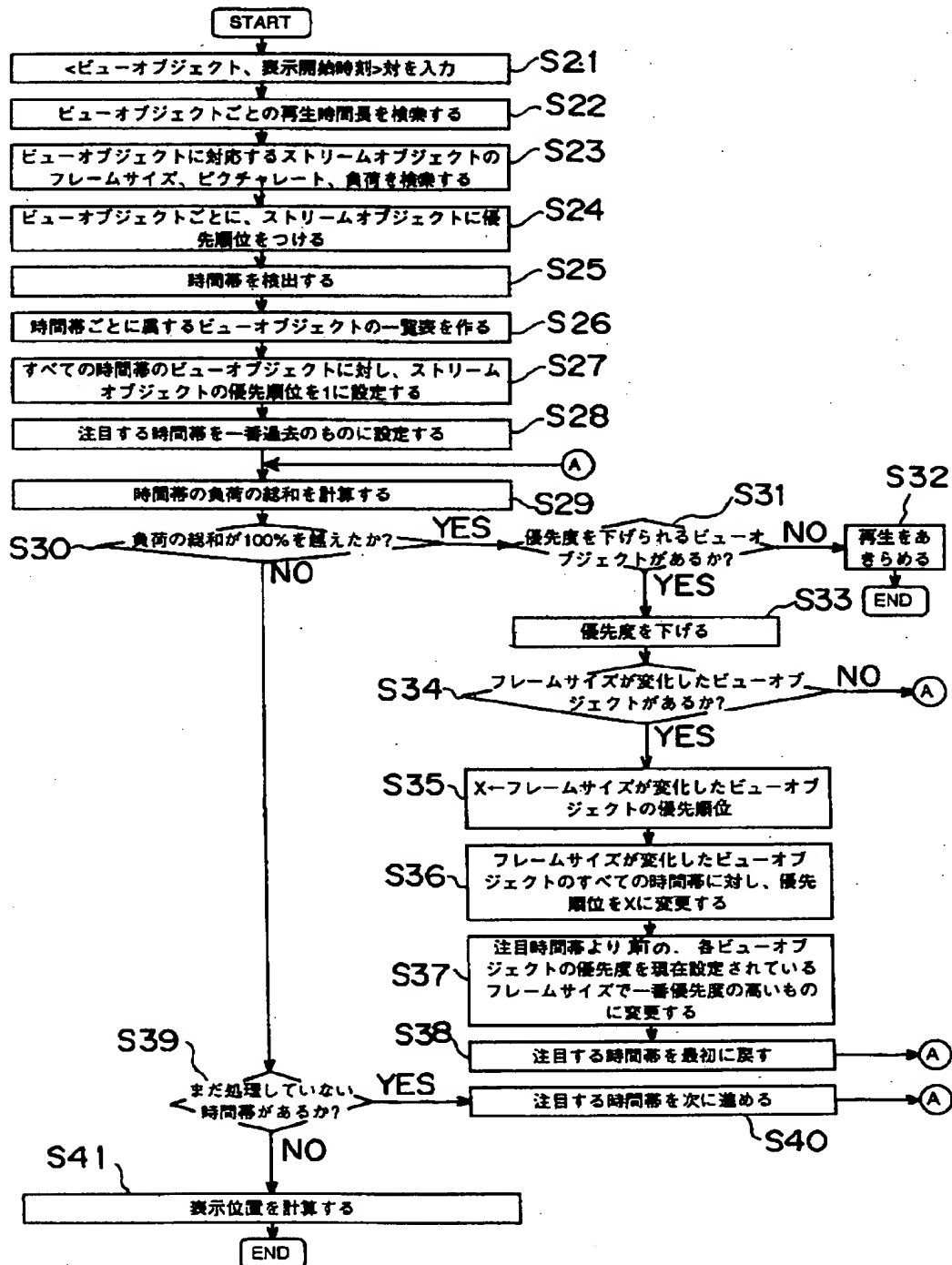
【図 1 4】

本発明の一実施例による
ビューオブジェクトの再生方法を示すフローチャート



【図 16】

本発明の一実施例によるスケジューラの動作方法を
示すフローチャート

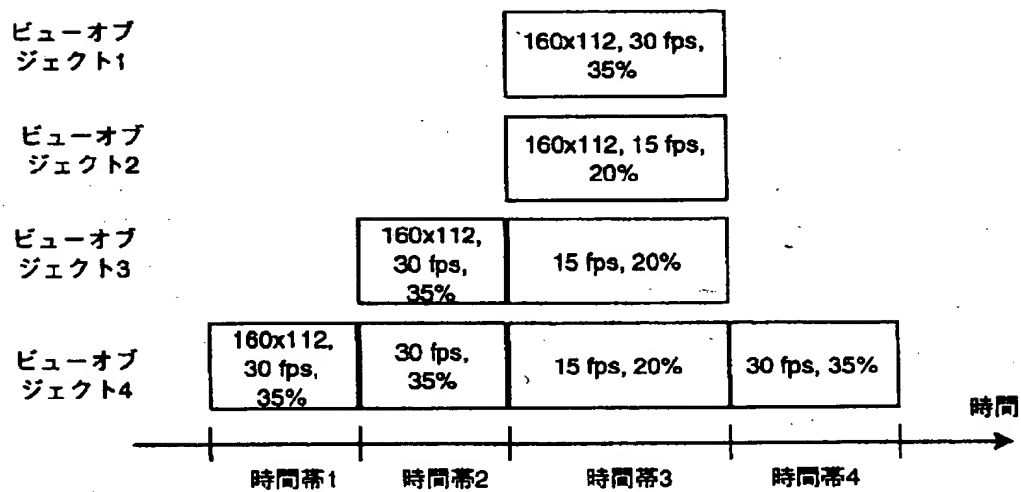


【図17】

本発明の一実施例によるスケジューラの
選択可能データとスケジュール結果とを示すフローチャート

優先順	フレームサイズ	fps	CPU負荷
1	320x240	30 fps	100 %
2	320x240	15 fps	60 %
3	320x240	10 fps	45 %
4	160x112	30 fps	35 %
5	160x112	15 fps	20 %
6	160x112	10 fps	15 %

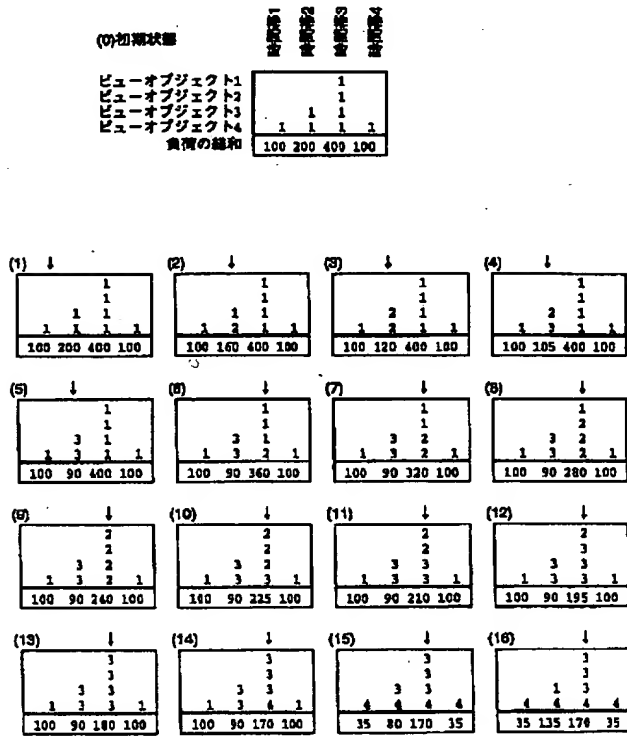
(a) 選択可能データ



(b) スケジュール結果

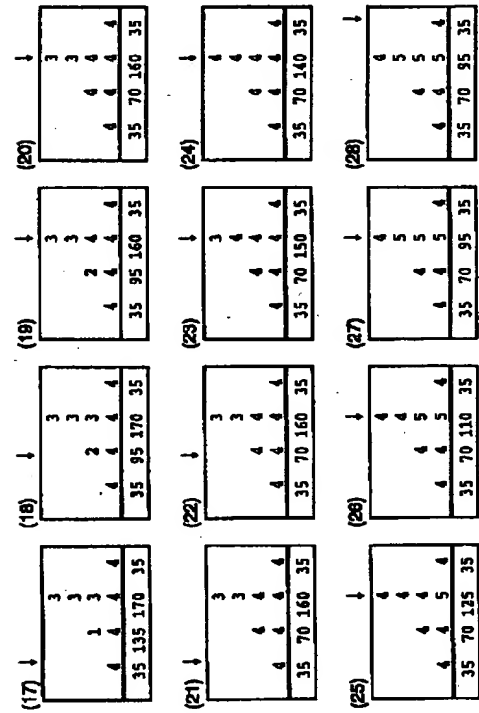
【図 18】

本発明の一実施例による
スケジューラの選択処理の経過を示す図



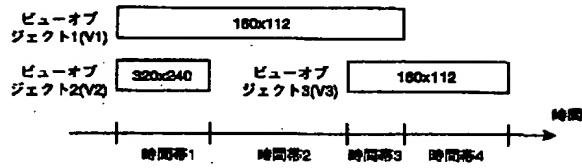
【図 19】

本発明の一実施例による
スケジューラの選択処理の経過を示す図

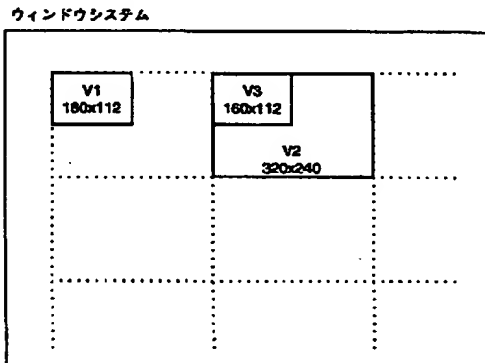


【図20】

本発明の一実施例によるスケジューラの
表示データの画面配置方法を示す図



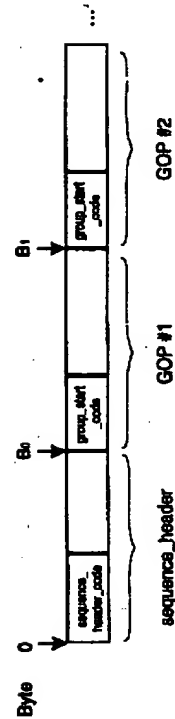
(a) データとフレームサイズ



(b) データのウィンドウへの割り当て

【図22】

本発明の一実施例によるマルチメディアデータ
検索システムのGOPテーブルの作成方法を示す図



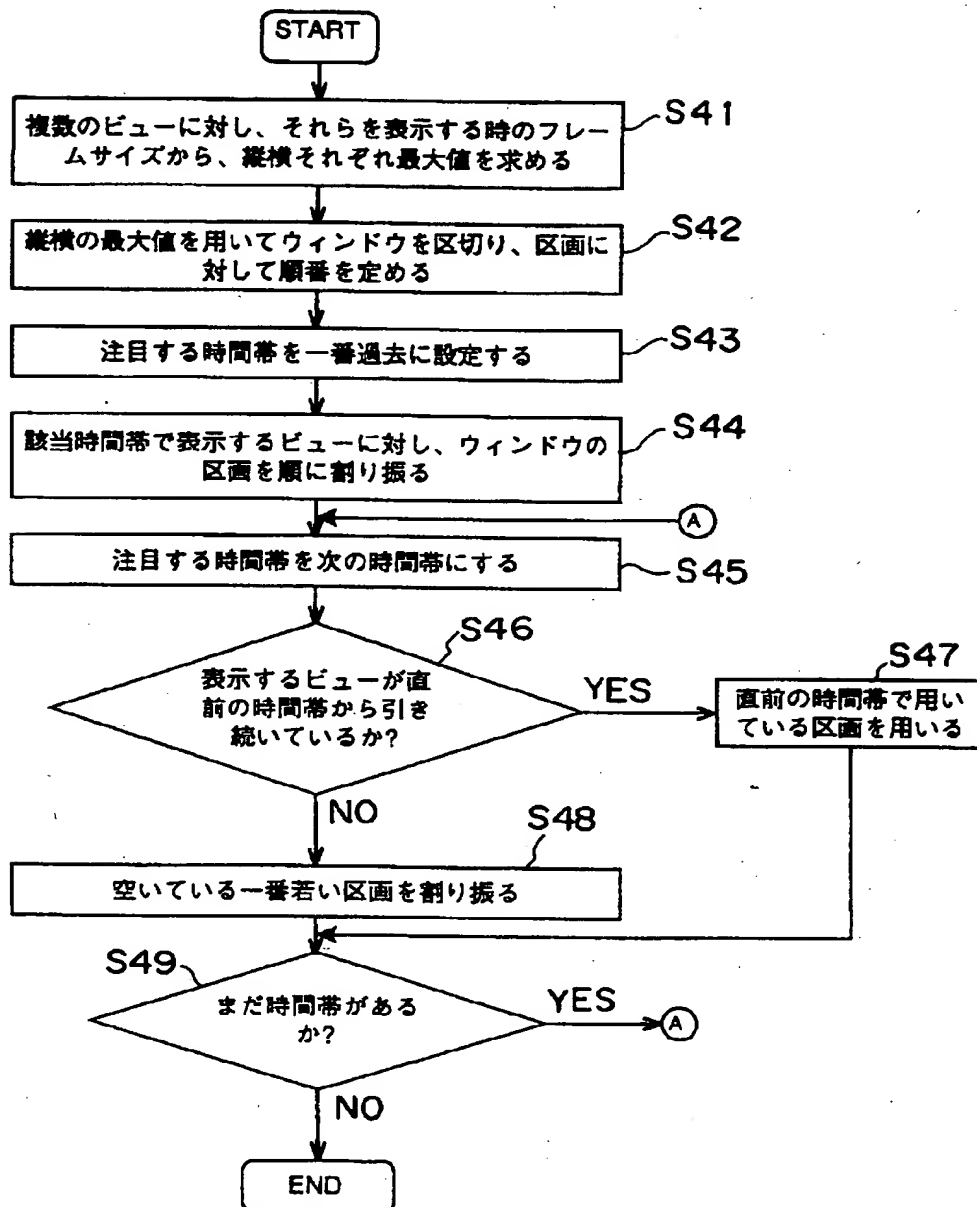
(a) MPEG-1 videoフォーマットの構成

frame #	byte
0	B0
7	B1
...	...

(b) GOPテーブル

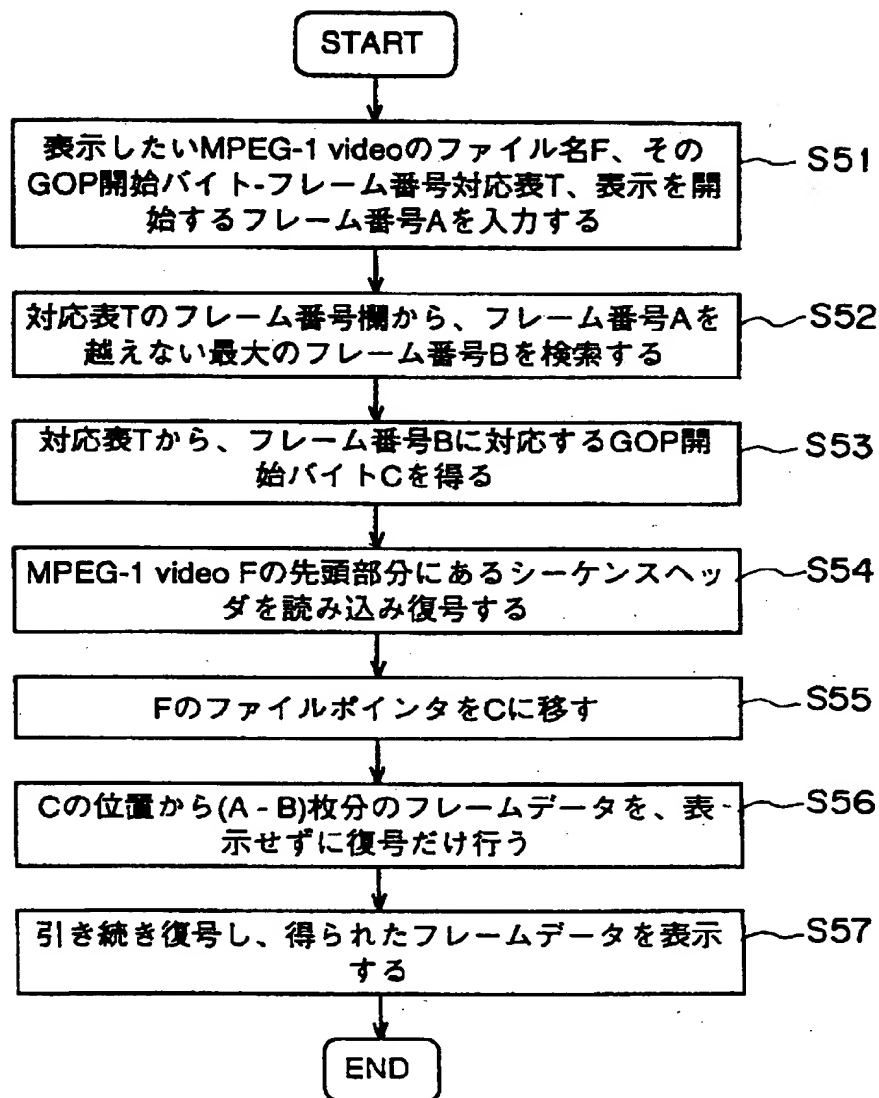
【図 21】

本発明の一実施例によるウィンドウ表示の
位置の決定方法を示すフローチャート



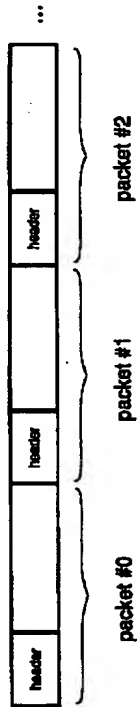
【図 23】

本発明の一実施例による MPEG-1 ビデオデータの
指定位置からの表示方法を示すフローチャート



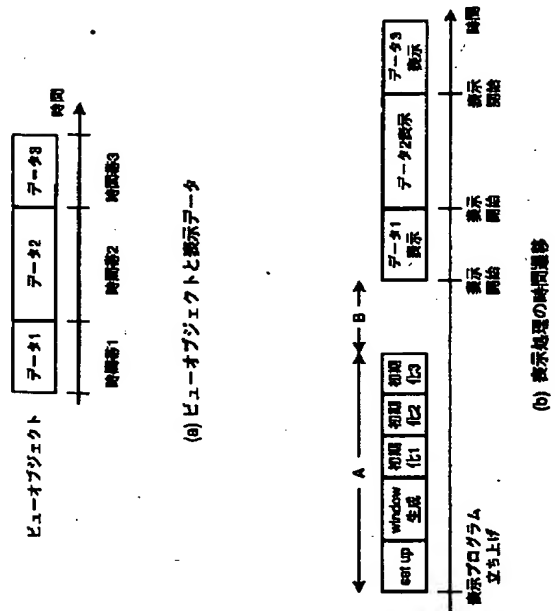
【図 24】

本発明の一実施例による MPEG-1
オーディオフォーマットの構成を示す図



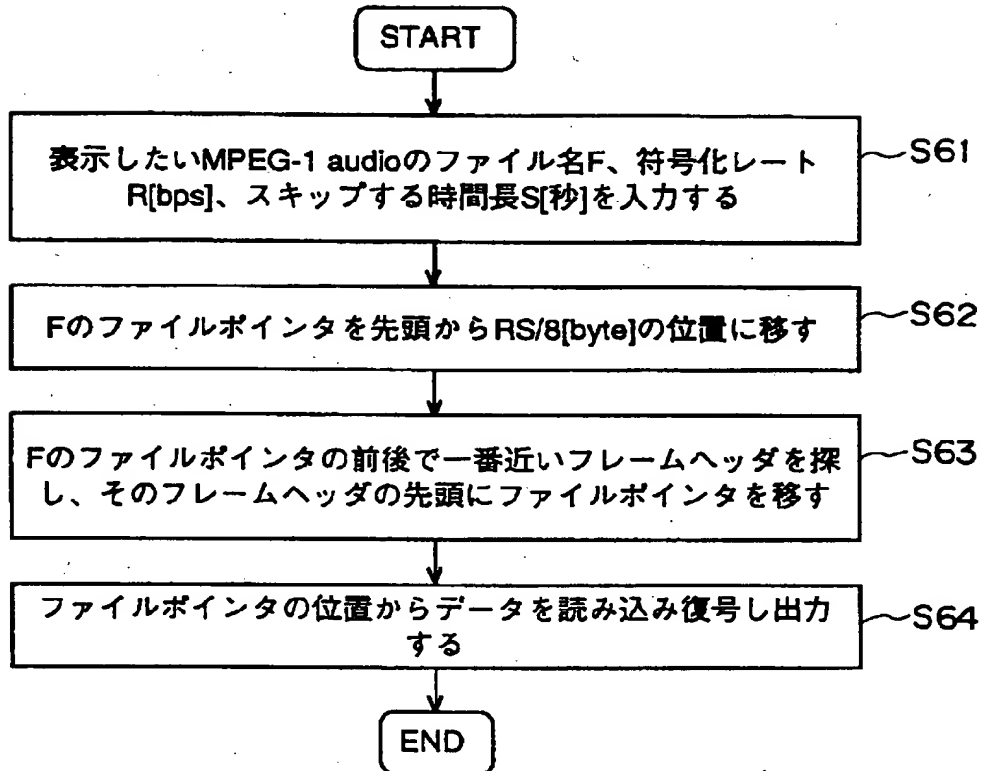
【図 26】

本発明の一実施例によるマルチメディアデータの
表示処理の時間遷移を示す図



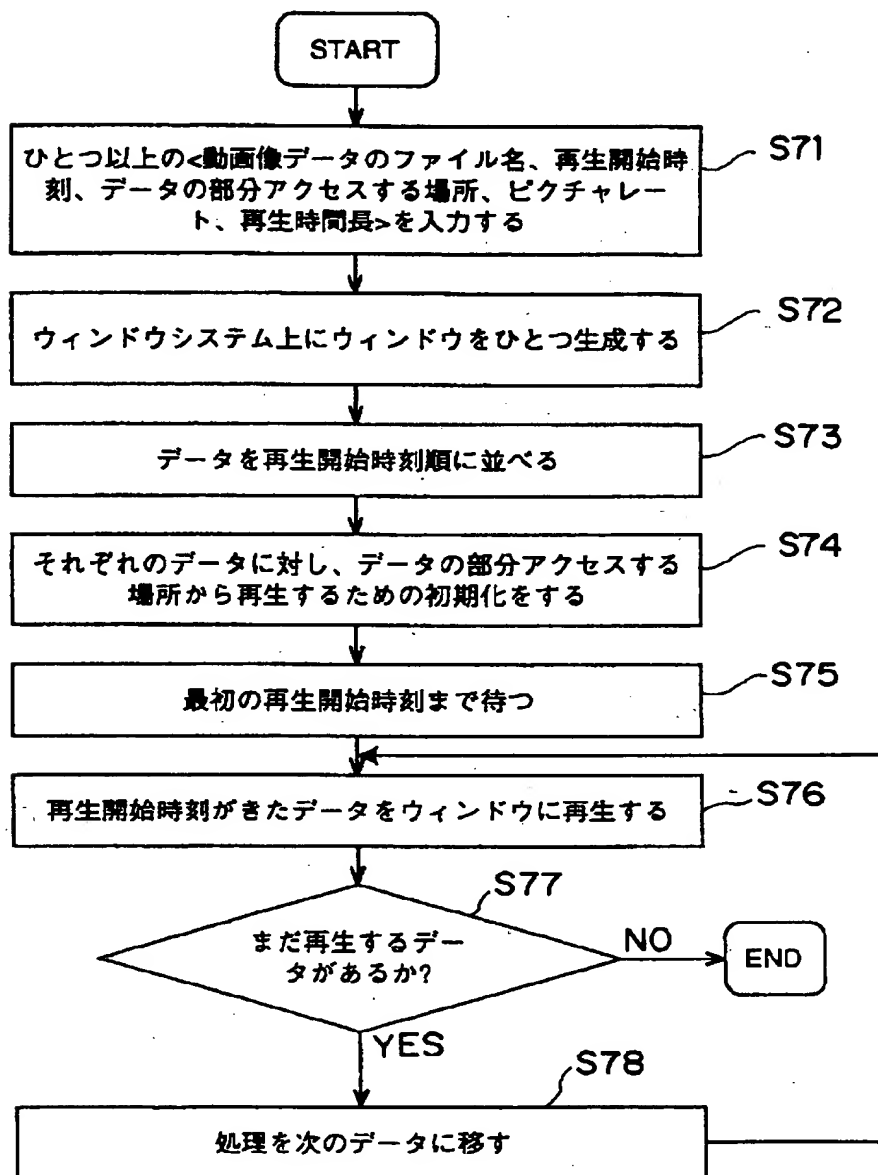
【図 25】

本発明の一実施例による MPEG-1
オーディオデータの指定位置からの表示方法を示す
フローチャート



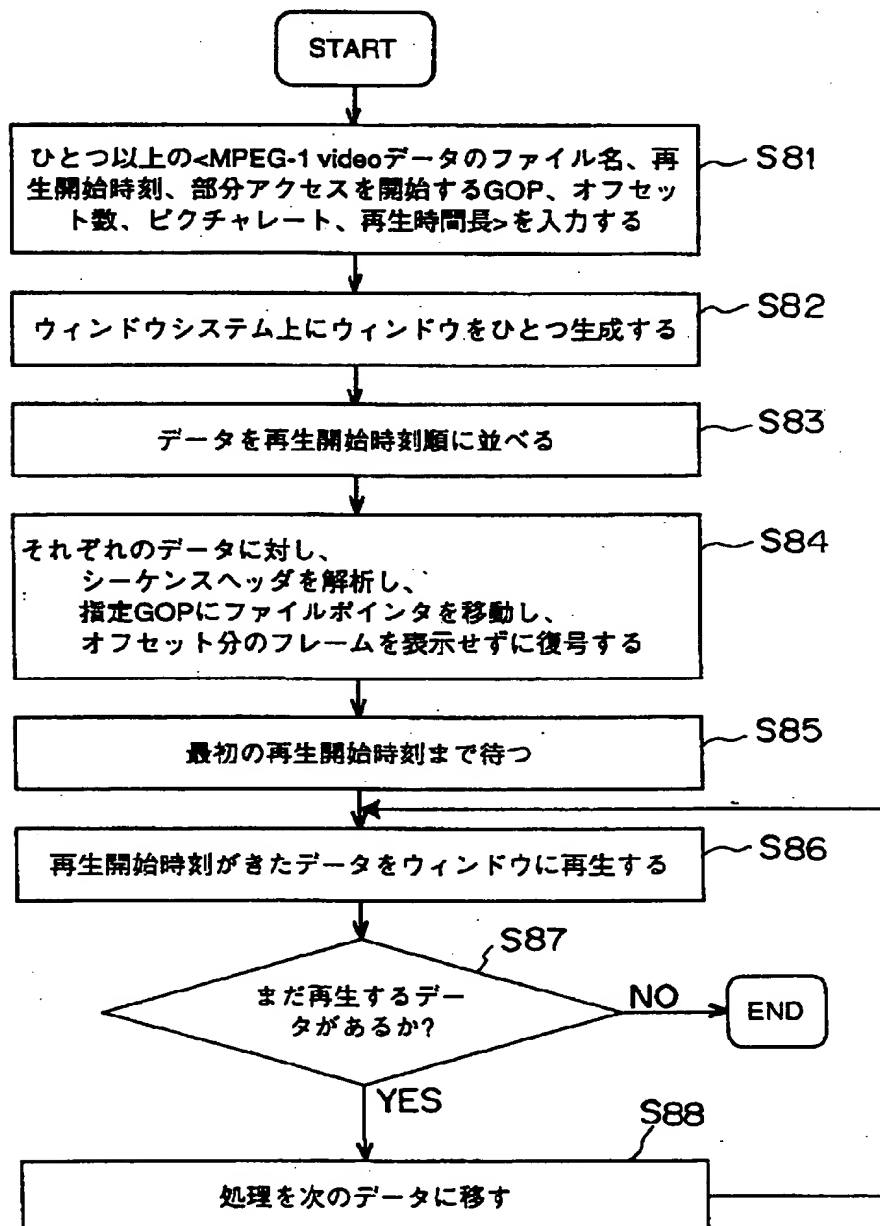
【図27】

本発明の一実施例によるMPEG-1ビデオデータの
同一ウィンドウに対する連続表示方法を示すフローチャート



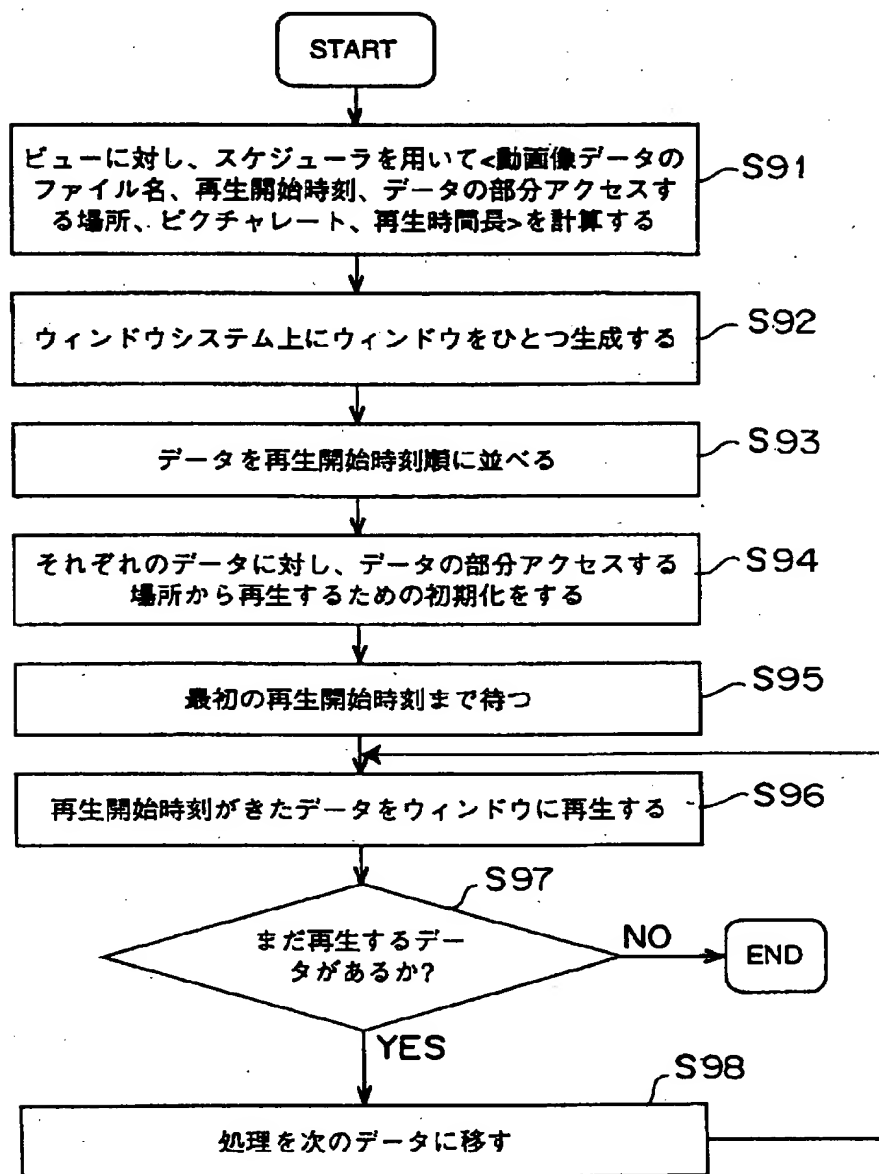
【図28】

本発明の一実施例によるMPEG-1ビデオデータの
同一ウィンドウに対する連続表示方法を示すフローチャート



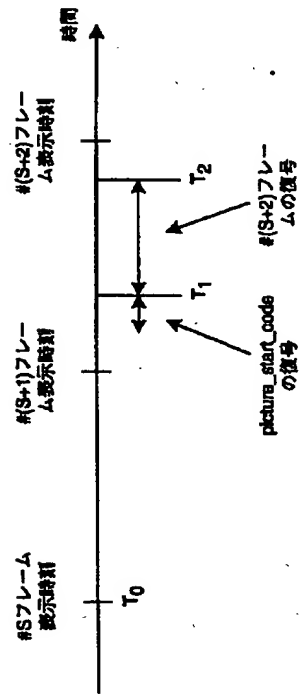
【図 29】

本発明の一実施例によるビューの複数データを用いる連続表示方法を示すフローチャート



【図 30】

本発明の一実施例によるフレーム表示タイミングを示す図



【図31】

本発明の一実施例によるMPEG-1ビデオデータのフレーム
表示方法を示すフローチャート

